Chips for a new wireless world™

DM256

Development Guide

Document no. 256-77-01 Draft L

1375 autoroute Transcanadienn
Bureau 30
Dorval (Québe
H9P 2W8 Canad

(514) 684-020
Fax: (514) 684-028

DM256

Development Guide

Document no. 256-77-01 Draft L

Wavesat Wireless Inc

1375 Trans Canada Highway, Suite 300

Dorval, Quebec

Canada, H9P 2W8

(888) 802-1616

(514) 684-0200

(514) 684-0288 (Fax)

www.wavesat.com

sales@wavesat.com

support@wavesat.com

Publication History

The publication history is given in the table below.

Release	Owner	Date	Description
A-C	L.Law		
D	L.Law	May 14, 2004	
E	M.Lee	June 28, 2004	General format and editorial changes.
F	M.Lee	July 30, 2004	Updating document to included the latest information concerning new load.
G	M.Lee	August 17, 2004	Update read and write commands lists. General format and editorial changes. Some of the commands are missing detailed procedures within the Development Guide. More procedures to come in future updates.
H	M.Lee	August 27, 2004	Adding more procedures for new commands. Updated BER test procedures to include test for both uncoded and coded.
I	M.Lee	September 10, 2004	General editorial updates, updates to command lists. Add a new column to the Master Command List to provide clarity on startup values vs recommended values.
J	M.Lee	September 24, 2004	General updates from Internal review for clarity and corrections. Updated list of commands to indicate whether commands are for use on the ASIC and FPGA or the ASIC only.
K	M.Lee	October 8, 2004	General updates and editorial changes from internal reviews.
L	M.Lee	November 1, 2004	More general updates and editorial changes from internal reviews. Add Base and Subscriber Station Scripts to Appendix F. Added new section for Analog Control Commands. Update uncoded BER Test procedures.

About Wavesat

Wavesat, founded in 1993 and headquartered in Montreal, is a leading supplier of wireless chipsets for the Broadband Wireless Access (BWA) industry. As a technology leader in NLOS BWA solutions, (Non Line of Sight) Wavesat is partnering with equipment manufacturers and system integrators to deliver best-in-class 802.16 compliant broadband systems based on the most advanced COFDM (Coded Orthogonal Frequency Division Multiplexing) developments.

The Wavesat offer includes a full suite of building blocks (including PHY, MAC hardware, processors, DACs, ADCs and other integrated elements) to build a wide range of distinctive and cost-effective BWA products, from highly efficient base stations to low power mobile CPE (Customer Premise Equipment).

The Wavesat technology is user friendly, easy to integrate and enables superior infrastructure and communication performances at lower cost than proprietary based solutions.

Wavesat leadership

At its asset and at the industry's benefit, Wavesat is the first in the industry to introduce an IEEE 802.16-2004 compliant PHY chipset with forward compatibility to 802.16e standard for mobile applications.

As a premier communications integrated circuit provider, Wavesat has unparalleled ability to offer BWA system design solutions, which are supported by user friendly development system, comprehensive reference designs and resourceful talented field engineering teams. All these valuable design tools, customer support services and engineering competencies are available to Wavesat system partners.

Wavesat, a skilled and dedicated team

Wavesat's dynamic and diverse team of scientists and engineers is entirely committed to delivering the highest quality and most cost-effective broadband wireless products available on the market. This dedication and passion has resulted in Wavesat producing innovative wireless solutions which push the boundary on what is currently possible. In many ways, the Wavesat R&D team is a "think tank" for wireless developments, especially in the leading edge field of OFDM technology.

Wavesat, more than 10 years of commitment to the wireless industry

The company has built a solid background and reputation in delivering baseband, IF and RF subsystems for the wireless industry. Wavesat now dedicates all its engineering and marketing resources to the IEEE standard based BWA industry as a fabless BWA semiconductor company.

Wavesat Wireless Inc.

1375 Trans Canada Highway, Suite 300

Dorval, Quebec

Canada, H9P 2W8

(888) 802-1616

(514) 684-0200

(514) 684-0288 (Fax)

www.wavesat.com

sales@wavesat.com

support@wavesat.com

This page has been left intentionally blank.

List of Sections

Appendix F: Development Kit Booting Scripts

Table of Contents

List of Figures

This page has been left intentionally blank.

List of Tables

This page has been left intentionally blank.

Section: 1 About This Document

1.1 Objective

The objective of this document is to provide the necessary information to:

- Learn about the DM256 chip's functionality,
- Evaluate the DM256 chip,
- Test the DM256 chip.

1.2 Scope

This document covers the following:

- Software load 1.32
- Dragon digital board.
- Anaconda analog board.

1.3 Document Information Guide

- **Note or Caution boxes** will be shown in the document as in the below example:

Note:	Note and Caution boxes contain important information. Please ensure that they are read and understood before proceeding to the next step.

These Note or Caution boxes contain important information concerning information provided in the previous stated information.

- When **information/commands** need to be typed into the console/computer these will be shown as in the below example by using font type Courier New 10 pts using font color Gray at 50:

```
Need to enter this information to the console ↵
```

- When the **Enter key** on your keyboard is required to terminate a command this will be shown in the document as in the below example:

↵

Note:	The Enter key is noted by the following symbol: ↵

- When the **console/computer returns information/messages** to the screen this will be shown in the document as in the below example by using font type Courier New **Bold** *Italic* 10 pts using font color Gray at 50:

Anaconda control version: x.xx

- When following procedure steps in the document ensure that you follow the procedure until you find the **end sign** at the bottom of the procedure as found below.

-End-

- Otherwise you will find a **continue sign** towards the end of your page asking you to continue on to the next page for the rest of the procedure.

-Continue-

- **Where parameter and description boxes** shown below are used to indicate for a particular parameter which are the possible values which can be inputted/assigned to the preceding command.

```
./phy_ctrl set tx_data_fifo_sync_reset value ↵
```

Where

Parameter	Description
value	0: Normal setting 1: The buffer is cleared

- **Commands which have [ ] (square brackets), these indicate optional parmeters. Otherwise the parameter is mandatory.**

```
./ofdm_stats data [cid] ↵
```

```
./phy_ctrl set tx_data_fifo_sync_reset value ↵
```

- **Commands which have | (pipe), they indicate that either of the parameters can be selected**

```
./ofdm_stats reset all|cid cid_value ↵
```

- **Commands which have value, they indicate that a parameter value (specified in the text) must be entered**

Set the Tx gain to a value between 0 and 255 by typing the following command:

```
./phy_ctrl set tx_gain_value value↵
```

This page has been left intentionally blank.

Section: 2 Description

2.1 Hardware Interface and Connections

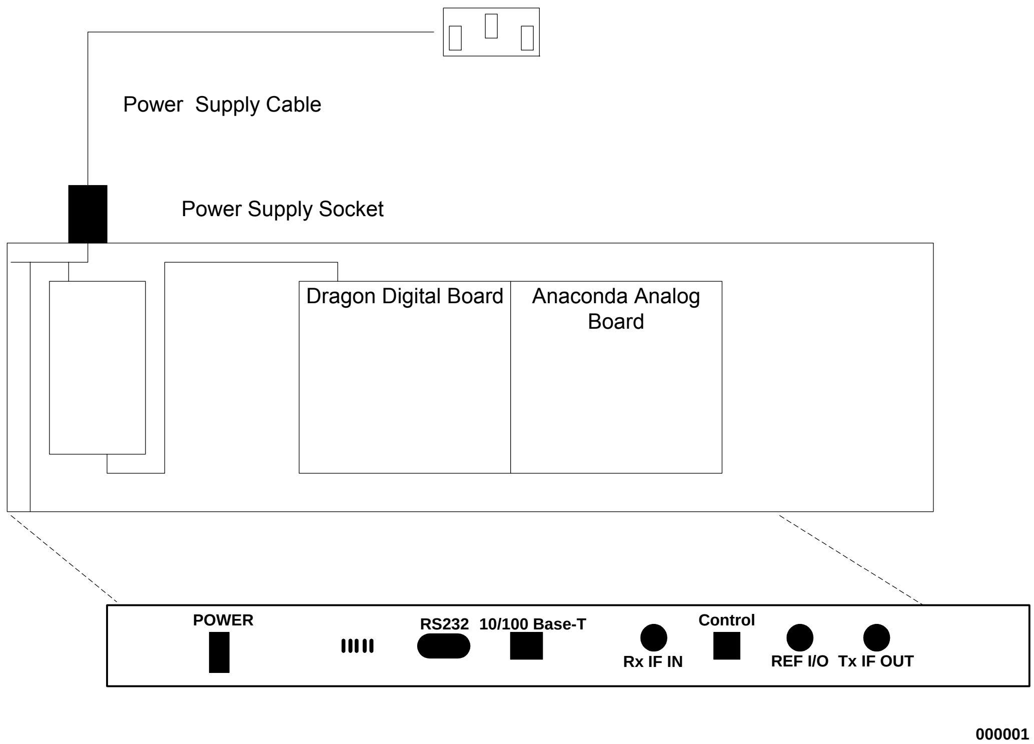


Figure 1 - Power Supply

2.1.1 Power Supply

A power cable is provided with the development kit and connects to the power supply socket located on the left rear side of the development kit.

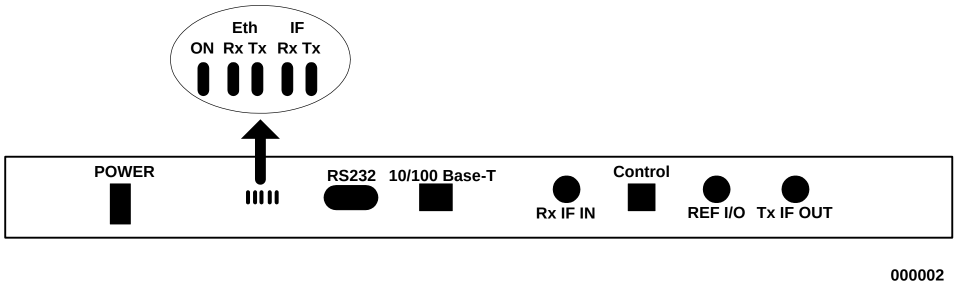


Figure 2 - Console Interface

2.1.2 Power Switch

The power switch is located on the left side of the console interface. It turns the development kit On or Off. and can also be used to reset the development kit.

2.1.3 LEDs

The five LEDs on the front face of the development kit serve the following functions:

Table 1 - LED Definition on Console Interface

Color	Function
Green	“Power” LED
Yellow	“Ethernet Receive” LED, controlled by Ethernet PHY chip
Green	“Ethernet Transmit” LED, controlled by Ethernet PHY chip
Yellow	“RF RX” LED, controlled by DM256
Green	“RF TX” LED, controlled by DM256

2.1.4 RS232 Port

The female DB9 port serves to connect a terminal or a computer configured as a terminal emulator. The pinouts of the port are as follows:

Table 2 - Pinouts of Female DB9 Port

Signal name	In / Out	Pin #	Description
TXD	Out	2	RS232 transmit data
RXD	In	3	RS232 receive data
GND		5	

2.1.5 10/100 Base-T Port

The RJ45 Ethernet port is used in Network File System (NFS) mode only and serves to download the FPGA (Field Programmable Gate Array) load to the FPGA chip. These pinouts are as follows:

Table 3 - RJ45 Ethernet Port Pinouts

Pin #	Signal name
1	TX+
2	TX-
3	RX+
4	
5	
6	RX-
7	
8	

2.1.6 Tx IF IN / Rx IF OUT Ports

A SMA connector is required for each of the following ports:

- Rx IF IN port
- Tx IF OUT port.

2.1.7 REF I/O

This SMA port can be used to connect the development kit to an external clock instead of using the built-in clock.

2.1.8 Control

This port is currently not used.

2.2 Typical Equipment Setup

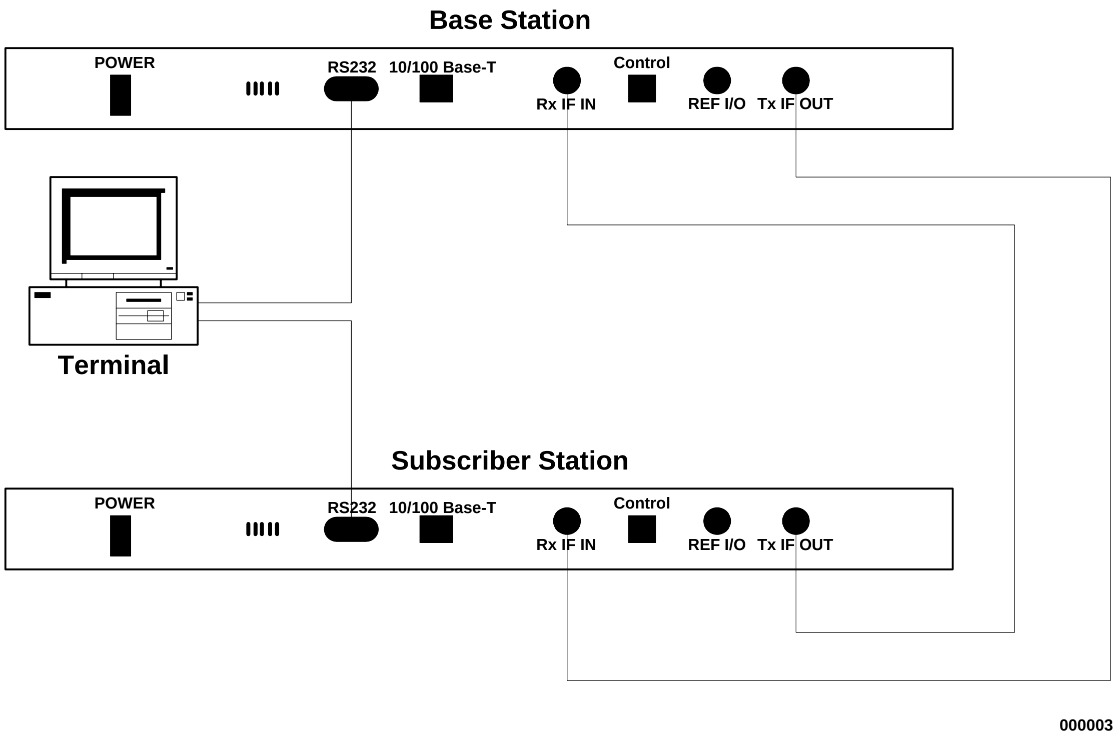


Figure 3 - Typical Equipment Setup

2.3 Memory Block Diagram

The software is stored in FLASH memory (8 MB) and running from SDRAM (16 MB). There is also a boot flash memory (512 KB) and a configuration EEPROM (512 Bytes).

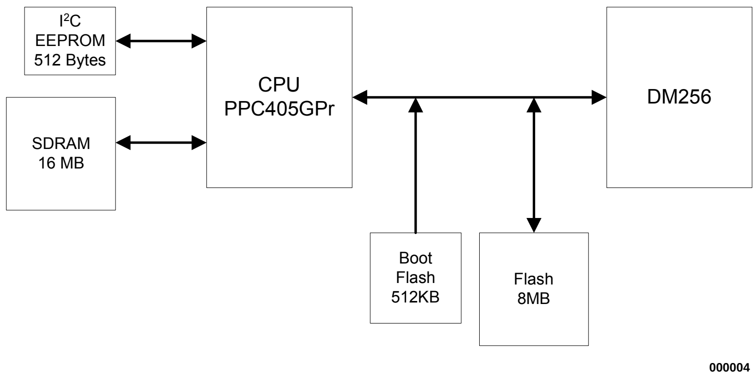


Figure 4 - Memory Block Diagram

Table 4 - Memory Map for Development Kit

Address Field (Start/Stop)	CS	PPC405 Control Machine	Memory Size	Description	Port size
0x00000000 0x03FFFFFF	-	SDRAM	64M	SDRAM, Application code and data	32 bits
0x04000000 0x07FFFFFF	-	-	64M	Reserved for SDRAM Memory expansion	-
0x08000000 0x087FFFFF	CS1	EBC	8M	FLASH memory	16 bits
0x08800000 0x08FFFFFF	-	-	8M	Reserved for Flash Memory expansion	-
0x09000000 0x7FFFFFFF	-	-	1.9G	Unused	-
0x80000000 0xEF5FFFFF	-	PCI	-	Reserved for PCI devices	-
0xEF600000 0xEFFFFFFF	-	-	10M	Internal Peripherals	-
0xF0000000 0xFFDFFFFF	-	-	260M	Unused	-
0xFFE00000 0xFFFFFFFF	CSO	EBC	2M	Boot Flash	8 bits

This page has been left intentionally blank.

Section: 3 Quick Startup

The quick start up section provides the necessary information to users who wish to:

- Connect development kits to your computer.
- Configure your computer as a terminal.
- Power up the development kits.
- Configure the development kits with scripts.
- Modify the configuration scripts.

3.1 Connect the Development kits to your Computer

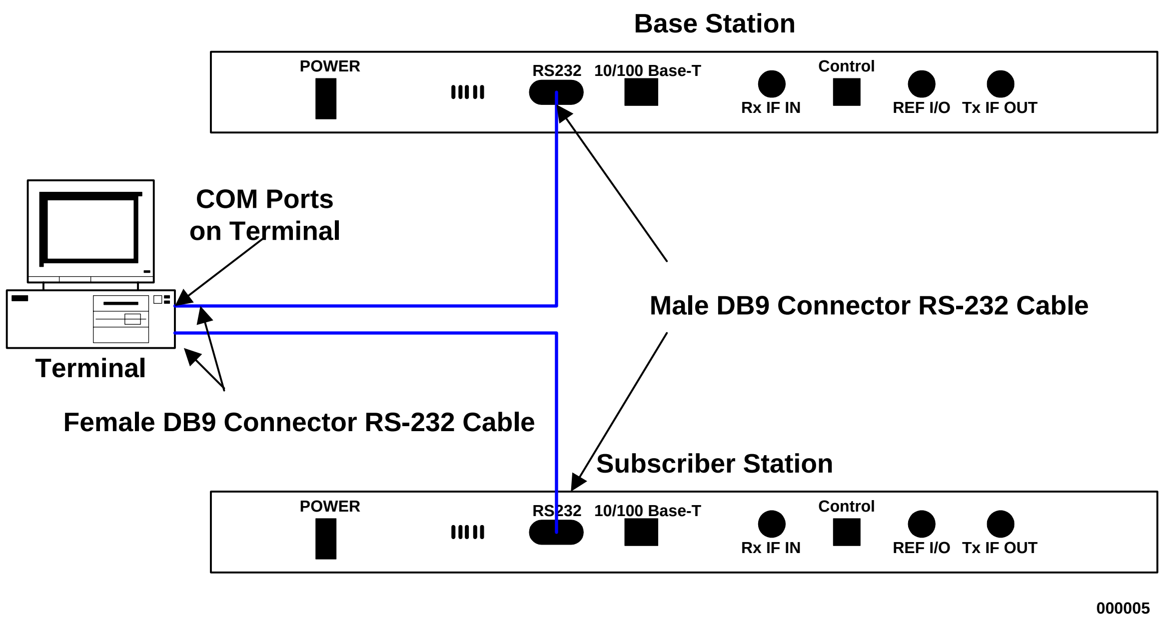


Figure 5 - Terminal to Development kit RS-232 Cable Connection

To connect each development kit to your computer, proceed as follows:

1. Use a straight RS-232 cable with a male DB9 connector at one end and a female DB9 connector at the other end (not null modem or crossover cables). See Appendix E: RS-232 Cable for more information about this cable.
2. Connect the male DB9 connector to the RS-232 port on the development kit.
3. Connect the female DB9 connector to a COM port of your computer.

-End-

3.2 Configure your Computer as a Terminal

From your computer, open a terminal emulator such as ***Minicom*** or ***HyperTerminal*** and configure the COM port as follows:

Table 5 - COM Port Configuration for Terminal Emulator

Baud Rate	9600
Data Bits	8
Parity	None
Stop Bits	1
Flow Control	None
Emulation	VT100

3.3 Power up the Development kits

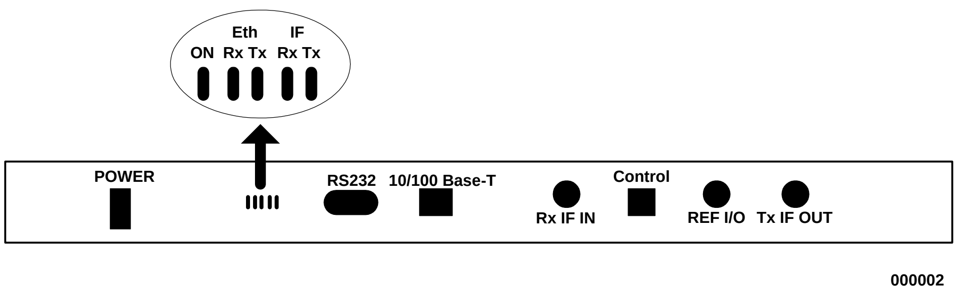


Figure 6 - Console Interface

To turn each development kit On, proceed as follows:

1. Turn the development kit On by toggling the POWER switch on the faceplate of the development kit.

   The ON LED will illuminate.

2. Once the booting process is completed and the kernel is loaded from the memory flash to the DM256 chip, the following message appears:

   Please press the Enter to activate this console.

3. Press the Enter key on your keyboard to activate your serial connection.

   ↵

Note:	The Enter key is noted by the following symbol: ↵

   When the # symbol appears, you are ready to start.

   #

-End-

3.4 Configure the Development kits with Scripts

The development kit comes with two configuration scripts to enable you to quickly and easily set up your two development kits. Not only the scripts avoid you from typing a long series of commands for configuring your development kits, they also contain typical or recommended settings. The script for configuring a development kit as a base station is usually identified as “base” and the one for configuring a development kit as a subscriber station is usually identified as “sub”.

3.4.1 Configuring the Base Station

Configure one development kit as the base station proceed as follows:

1. Go to the dragon directory by typing:

   `cd /dragon ↵`

2. Execute the base station script by typing:

   `./base ↵`

-End-

3.4.2 Configuring the Subscriber Station

Configure the other development kit as the subscriber station proceed as follows:

1. Go to the dragon directory by typing:

   `cd /dragon ↵`

2. Execute the subscriber station script by typing:

   `./sub ↵`

-End-

3.5 Modify the Configuration Scripts

You may need to modify one or both configuration scripts that have been provided with your development kit under either of the following conditions:

- You need to modify some configuration parameter settings.
- You have upgraded the FPGA file.

3.5.1 View Script Contents

To view the contents of a script, proceed as follows:

1. Go to the dragon directory by typing:

   `cd /dragon ↵`

2. Display the contents of the script by typing the following:

   `cat base ↵` : for the base station script

   `cat sub ↵` : for the subscriber station script

-End-

3.5.2 Modify Script Contents

To modify the contents of a script, proceed as follows:

1. Go to the dragon directory by typing:

   `cd /dragon ↵`

2. Open the script using the vi editor by typing the following:

   `vi base ↵` : for the base station script

   `vi sub ↵` : for the subscriber station script

3. Modify the contents of the script using the vi editor. (See Appendix A for the list of vi commands.)

-End-

Section: 4 Quick-Reference Section

4.1 Verify the Versions and Revisions

Use this section to verify versions or revisions of the following:

- FPGA Load Version.
- DM256 Software Version.
- Scheduler Version.
- Analog Software Version.
- Analog Board Revision.
- Statistics Software Version.

Note:	All of the below commands return the same value, the software versions. Currently the code for the development kit is delivered in one software load instead of individual loads for each component.

4.1.1 FPGA Load Version

To verify the version of the FPGA load that is currently in the flash memory, type:

```
./fpga_prog version ↵
```

Note:	The Enter key is noted by the following symbol: ↵

The console displays the results as follows:

```
FPGA prog version: x.xx
```

4.1.2 DM256 Software Version

To verify the DM256 version, type:

```
./phy_ctrl version ↵
```

The console displays the results as follows:

```
PHY driver version: x.xx
PHY control version: x.xx
```

4.1.3 Scheduler Version

This command is performed at the base station development kit. To obtain the version of the Scheduler Control, type:

```
./sched_ctrl version ↵
```

The console displays the results as follows:

```
Scheduler control version: x.xx
```

4.1.4 Analog Software Version

To obtain the analog software version, type:

```
./anaconda_ctrl version ↵
```

The console displays the results as follows:

```
Anaconda control version: x.xx
```

4.1.5 Analog Board Revision

To obtain the analog board revision type:

```
./anaconda_ctrl get revision ↵
```

The console displays the results as follows:

```
revision = x
```

Note:	The console displays a numeric value that must be translated to the board revision as follows: 1=A, 2=B, 3=C etc.

4.1.6 Statistics Software Version

To obtain the version of the statistics software, type:

```
./ofdm_stats version ↵
```

The console displays the results as follows:

```
Getting version
OFDM network driver version: x.xx
OFDM stats version: x.xx
```

4.2 Dumping of the DM 256 Register

Use this section to obtain a dump the of the DM 256 register:

To display the contents of the DM 256 register contents, type:

```
./phy_ctrl dump ↵
```

The console displays the results similar to as follows:

```
Registers dump for OFDM PHY
-------------------------------
          status 0x00000008
         control 0x03020083
        revision 0x57030044
       interrupt 0x000001FC
            dma1 0xC0000000
            dma2 0x000B4000
     conf_symbol 0x00000011
             sid 0x00000009
      sync_force 0x00000004
     sync_thresh 0x3FBC7D00
      delay_corr 0x00000000
        lost_pak 0x00000000
        frame_nb 0x002311D9
          config 0x00000004
    config_value 0x000000FA
      digital_if 0x00000223
     digital_if2 0x40004000
            papr 0x0000000F
   tx_mute_delay 0x0000043C
         cp_size 0x00000610
             spi 0x46900203
             dac 0x00500050
            dac2 0x00000000
  analog_control 0x00400000
    analog_value 0x00000000
       snr_power 0x00000000
```

4.3 Display the list of commands

Use this section to display the list of get, set and/or user commands for the following:

- anaconda_ctrl.
- phy_ctrl.
- sched_ctrl.

4.3.1 anaconda_ctrl get and set Command Lists

To display a command list of get commands under anaconda_ctrl, type:

```
./anaconda_ctrl print_get ↵
```

The console displays the results as follows:

```
List of set parameters:
revision
test_mode
sample_clk
```

To display a command list of set commands under anaconda_ctrl, type:

```
./anaconda_ctrl print_set ↵
```

The console displays the results as follows:

```
List of set parameters:
test_mode
sample_clk
```

4.3.2 phy_ctrl get and set Command Lists

To display a command list of get commands under phy_ctrl, type:

```
./phy_ctrl print_get ↵
```

The console displays the results as follows:

```
fifo register:
rx_fifo -

status register:
tx_data_ready - rx_data_ready - rx_link_config -
rx_link_active -
```

Note:	This is only a partial list of the commands.

-To display a command list of set commands under phy_ctrl, type:

```
./phy_ctrl print_set ↵
```

The console displays the results as follows:

```
fifo register:
tx_fifo -

interrupt register:
tx_fifo_underrun - int_header_err - cfg_symb_err -
tx_fifo_reset - config_symb - tx_frame -
rx_buff_overflow - rx_frame - tx_data_ready -
rx_data_ready - reset_irq_stats –
```

Note:	This is only a partial list of the commands.

4.3.3 sched_ctrl get, set and user Command Lists

To display a command list of get commands under sched_ctrl, type:

```
./sched_ctrl print_get ↵
```

The console displays the results as follows:

```
List of get parameters:
status
default_rate_id
dcd_change_count
test_stc_zone
pause
debug
bert
duplex_mode
feedback_mode
operating_mode
gap
ttg
rtg
FEC
contention_slot
cp_size
channel_bw
frame_code
```

To display a command list of set commands under sched_ctrl, type:

```
./sched_ctrl print_set ↵
```

The console displays the results as follows:

```
List of set parameters:
pause
start
debug
bert
duplex_mode
feedback_mode
operating_mode
gap
dcd_change_count
test_stc_zone
ttg
rtg
FEC
contention_slot
cp_size
channel_bw
frame_code
```

To display a command list of user commands under sched_ctrl, type:

```
./sched_ctrl print_user ↵
```

The console displays the results as follows:

```
List of user parameters:
add
del
modify
print
```

This page has been left intentionally blank.

4.4 Read and Write Commands

The following table lists, in alphabetical order grouped in read and write commands, the commands which are included in your development kit.

Note: Some of commands listed below are not for customer use. See Table below for specifics.

For Customer Use = √
Command Not Used = X

Read commands = get = R
Write commands = set = W

Settings at Startup = values of registers, commands when the Development Kit is turned on.
Recommended Settings = Wavesat recommends that you change the startup settings to these settings.

For more detailed information about registers and bit information see DataBook Section on Register Descriptions.

Table 6 - Read and Write Commands

Read and Write Command Names All Commands for use with ASIC and FPGA Except * ASIC Commands Only	**Read or Write Command**	**Read and Write Command Descriptions**	**Registers Name**	**Settings at Startup**	**Recommended Setting**	**For Customer Use**	**For Wavesat Use**	**Where BS**	**Where SS**
./anaconda_ctrl get revision	R	Show the revision of the analog board.				√		√	√
./anaconda_ctrl get sample_clk	R	Show the sampling frequency in Hz.				√		√	√
./anaconda_ctrl set sample_clk value	W	Set the sampling frequency: **Value** = 13000000, 33333333, 40000000 or 66666666 Hz.		40000000		√		√	√
./anaconda_ctrl get test_mode	R	Wavesat use only.					√	√	√
./anaconda_ctrl set test_mode value	W	Wavesat use only.		0			√	√	√
./anaconda_ctrl print_get	R	Show all commands with ./anaconda_ctrl get.				√		√	√
./anaconda_ctrl print_set	R	Show all commands with ./anaconda_ctrl set.				√		√	√
./anaconda_ctrl version	R	Show the version of the analog software.				√		√	√
./fpga_prog version	R	Show the version of the FPGA load.				√		√	√
./fpga_prog [–c] fpga_load_file	W	To program the DM 256 with a compressed file. The –c flag indicates the file is stored in a compressed file format .gz.					√	√	√

Read and Write Command Names All Commands for use with ASIC and FPGA Except * ASIC Commands Only	**Read or Write Command**	**Read and Write Command Descriptions**	**Registers Name**	**Settings at Startup**	**Recommended Setting**	**For Customer Use**	**For Wavesat Use**	**Where BS**	**Where SS**
./macdb_ctrl add MAC_ADDRESS CID Type DL_Mod #DL_Sym UL_Mod #UL_Sym Preamble Midamble	W	Add a user connection using the following parameters: • MAC ADDRESS: Subscriber Station MAC Address. • CID: connection ID (1 – 36). • Type: Transport or LinkTest. • DL_Mod: Modulation rate to use in downlink (BPSK-1/2, QPSK-1/2, QPSK-3/4, QAM16-1/2, QAM16-3/4, QAM64-2/3, QAM 64-3/4, BPSK-3/4). • #DL_Sym: Number of data symbols to allocate in downlink. • UL_Mod: Modulation rate to use in uplink (BPSK-1/2, QPSK-1/2, QPSK-3/4, QAM16-1/2, QAM16-3/4, QAM64-2/3, QAM 64-3/4, BPSK-3/4). • #UL_Sym: Number of data symbols to allocate in uplink. • Preamble (0 or 1). • Midamble (0-3).				√		√	
./macdb_ctrl del MAC|CID arg	W	Delete a connection using the connection ID (1 – 36) or MAC Address (Subscriber Station MAC Address).				√		√	
./macdb_ctrl modify CID DL_Modulation #DL_Symbols UL_Modulation #UL_Symbols Preamble Midamble	W	Modify a user connection using the following parameters: • CID: connection ID (1 – 36). • DL_Modulation: New modulation rate to use in downlink (BPSK-1/2, QPSK-1/2, QPSK-3/4, QAM16-1/2, QAM16-3/4, QAM64-2/3, QAM 64-3/4, BPSK-3/4). • #DL_Symbols: New number of data symbols to allocate in downlink. • UL_Modulation: New modulation rate to use in uplink (BPSK-1/2, QPSK-1/2, QPSK-3/4, QAM16-1/2, QAM16-3/4, QAM64-2/3, QAM 64-3/4, BPSK-3/4). • #UL_Symbols: New number of data symbols to allocate in uplink. • Preamble (0 or 1). • Midamble (0-3).				√		√	
./macdb_ctrl show	R	Show the list of connections.				√		√	
./ofdm_stats ber cid	R	Show statistics on the BER for the specified connection.				√		√	√

Read and Write Command Names All Commands for use with ASIC and FPGA Except * ASIC Commands Only	**Read or Write Command**	**Read and Write Command Descriptions**	**Registers Name**	**Settings at Startup**	**Recommended Setting**	**For Customer Use**	**For Wavesat Use**	**Where BS**	**Where SS**
./ofdm_stats data cid	R	Show statistics on received packets for the specified CID connection.				√		√	√
./ofdm_stats link cid	R	Show statistics on the quality of the Rx link for the specified CID connection.				√		√	√
./ofdm_stats reset all | cid cid_value	W	Reset statistical data to 0 for: (all) all channels or (1-36) an individual CID channel or for an individual CID channel (1-36) to a set cid_value (0-37).				√		√	√
./ofdm_stats symbol REF1_CELL | CONFIG_CELL | IDLE_CELL | REF2_CELL | FCH_CELL | MIDAMBLE_CELL | GAP_CELL | DATA_CELL	R	Obtain link quality statistics on a particular symbol: REF1_CELL | CONFIG_CELL | IDLE_CELL | REF2_CELL | FCH_CELL | MIDAMBLE_CELL | GAP_CELL | DATA_CELL Certain symbol types are received only when the DM 256 is placed in certain modes: i.e. pdu_passthrough =1. REF1_CELL pdu_passthrough =1 REF2_CELL pdu_passthrough =1 FCH_CELL pdu_passthrough =1 MIDAMBLE_CELL pdu_passthrough =1 GAP_CELL pdu_passthrough =1 DATA_CELL pdu_passthrough =1 or 0 CONFIG_CELL, IDLE_CELL pdu_passthrough N/A.				√		√	√
./ofdm_stats version	R	Show the version of the stats module.				√		√	√
./phy_ctrl print_get	R	Show all commands with ./phy_ctrl get.				√		√	√
./phy_ctrl print_set	R	Show all commands with ./phy_ctrl set.				√		√	√
./phy_ctrl dump	R	Show the DM 256 register contents.					√	√	√
./phy_ctrl get afc	R	Show the contents of the AFC DAC register.	dac2			√			√
./phy_ctrl get afc_enable	R	Show whether of the AFC module is: (0) Disable or (1) Enabled.	dac2			√			√
./phy_ctrl set afc_enable value	W	Set the AFC module to: (0) Disable or (1) Enabled.	dac2	0		√			√
./phy_ctrl get agc	R	Show the contents of the AGC DAC register.	dac2			√			√
./phy_ctrl get agc_enable	R	Show whether of the AGC module is: (0) Disable or (1) Enabled.	dac2			√			√
./phy_ctrl set agc_enable value	W	Set the AGC module to: (0) Disable or (1) Enabled.	dac2	0		√			√
./phy_ctrl get agc_ctrl	R	Not used. Wavesat use only.				X	X	X	X

Read and Write Command Names All Commands for use with ASIC and FPGA Except * ASIC Commands Only	**Read or Write Command**	**Read and Write Command Descriptions**	**Registers Name**	**Settings at Startup**	**Recommended Setting**	**For Customer Use**	**For Wavesat Use**	**Where BS**	**Where SS**
./phy_ctrl get alc	R	Show the value for the ALC DAC register.	dac			X	X	X	X
./phy_ctrl set alc value	W	Set a value for the ALC DAC register (0-1023).	dac	80		√		√	√
./phy_ctrl get alc_enable	R	Show whether of the ALC module is: (0) Disable or (1) Enabled.	dac			X	X	X	X
./phy_ctrl set alc_enable value	W	Set the ALC module to: (0) Disable or (1) Enabled.	dac	0		√		v	√
./phy_ctrl get analog_adc_inphase *	R	Show the values obtained by inphase ADC.	analog_value			√		√	√
./phy_ctrl get analog_adc_mode *	R	Show the Analog ADC mode where: (0) Digital mode, Rx input are read from 10-bit RX_SAMPLES_PIN bus, Power Down ADC. (1) Analog mode, Internal ADC Standby. (2) Analog mode, Internal ADC using unsigned binary coding. (3) Analog mode, Internal ADC using 2's complement coding.	analog_control			√		√	√
./phy_ctrl set analog_adc_mode value *	W	To set the Analog ADC mode where: (0) Digital mode, Rx input are read from 10-bit RX_SAMPLES_PIN bus, Power Down ADC. (1) Analog mode, Internal ADC Standby. (2) Analog mode, Internal ADC using unsigned binary coding. (3) Analog mode, Internal ADC using 2's complement coding.	analog_control	0	2	√		√	√
./phy_ctrl get analog_adc_quad *	R	Show the values obtained by quadrature ADC.	analog_value			√		√	√
./phy_ctrl get analog_afc *	R	Show whether of the AFC DAC is: (0) Disable or (1) Enabled.	analog_control	0		√		√	√
./phy_ctrl set analog_afc value *	W	Set the AFC DAC module to: (0) Disable or (1) Enabled.	analog_control			√		√	√
./phy_ctrl get analog_agc *	R	Show whether of the AGC DAC is: (0) Disable or (1) Enabled.	analog_control	0		√		√	√

Read and Write Command Names All Commands for use with ASIC and FPGA Except * ASIC Commands Only	**Read or Write Command**	**Read and Write Command Descriptions**	**Registers Name**	**Settings at Startup**	**Recommended Setting**	**For Customer Use**	**For Wavesat Use**	**Where BS**	**Where SS**
./phy_ctrl set analog_agc value *	W	Set the AGC DAC module to: (0) Disable or (1) Enabled.	analog_control			√		√	√
./phy_ctrl get analog_alc *	R	Show whether of the ALC DAC is: (0) Disable or (1) Enabled.	analog_control			√		√	√
./phy_ctrl set analog_alc value *	W	Set the ALC DAC module to: (0) Disable or (1) Enabled.	analog_control	0		√		√	√
./phy_ctrl get analog_dac *	R	Show the value for all Analog DACs.	analog_value			√		√	√
./phy_ctrl set analog_dac value *	W	Set value for all Analog DACs when in Analog test mode.	analog_value			√		√	√
./phy_ctrl get analog_test *	R	Show whether of the Analog Test is: (0) Disable or (1) Enabled.	analog_value			√		√	√
./phy_ctrl set analog_test value *	W	Set the Analog Test to: (0) Disable, analog test: tx sample DAC come from Tx Digitial IF, AFC DAC=AFC module, AGC DAC=AGC module, ALC DAC = ALC in the bus interface or (1) Enabled, analog test: tx samples DAC, AFC DAC, AGC DAC, ALC DAC values all come from the Analog register in the bus interface or disable.	analog_value	0		√		√	√
./phy_ctrl get analog_test_pulse *	R	Show whether of the Analog Test Pulse is: (0) Disable or (1) Enabled.	analog_control			√		√	√
./phy_ctrl set analog_test_pulse value *	W	Set the Analog Test Pulse to: (0) Disable, Do not latch Analog value or (1) Enabled, Latch Analog value from Analog value register in the bus interface into AFC, AFC and ALC DAC.	analog_control	0		√		√	√
./phy_ctrl get analog_tx_inphase *	R	Show whether of the analog Tx DAC inphase DAC is: (0) Disable or (1) Enabled.	analog_control			√		√	√
./phy_ctrl set analog_tx_inphase value *	W	Set the analog Tx DAC inphase DAC to: (0) Disable or (1) Enabled.	analog_control	0		√		√	√
./phy_ctrl get analog_tx_samples_disable*	R	Show whether 10 bits digital tx samples pins are: (0) Disable, driven by the tx digital IF or (1) Enabled, driven to 0.	analog_control			√		√	√
./phy_ctrl set analog_tx_samples_disable value *	W	Set the 10 bits digital tx samples pins to be: (0) Disable, driven by the tx digital IF or (1) Enabled, driven to 0.	analog_control	0		√		√	√

Read and Write Command Names All Commands for use with ASIC and FPGA Except * ASIC Commands Only	**Read or Write Command**	**Read and Write Command Descriptions**	**Registers Name**	**Settings at Startup**	**Recommended Setting**	**For Customer Use**	**For Wavesat Use**	**Where BS**	**Where SS**
./phy_ctrl get analog_tx_quad *	R	Show whether analog Tx DAC quadrature DAC is: (0) Disable or (1) Enabled.	analog_control			√		√	√
./phy_ctrl set analog_tx_quad value *	W	Set the analog Tx DAC quadrature DAC: (0) Disable or (1) Enabled.	analog_control	0		√		√	√
./phy_ctrl get analog_tx_sleep *	R	Show whether sleep mode in DAC is: (0) Disable or (1) Enabled.	analog_control			√		√	√
./phy_ctrl set analog_tx_sleep value *	W	Set the sleep mode in DAC: (0) Disable or (1) Enabled.	analog_control	1		√		√	√
./phy_ctrl get antenna_switch	R	Show which antenna port is selected: (0) Port 0 or (1) Port 1.	tx_mute_delay			√		√	√
./phy_ctrl set antenna_switch value	W	Set the Switch antenna to: (0) Port 0 or (1) Port 1.	tx_mute_delay	0		√		√	√
./phy_ctrl get auto_tx_mute_disable	R	Show whether the auto Tx mute is: (0) Disable or (1) Enabled.	tx_mute_delay			√		√	√
./phy_ctrl set auto_tx_mute_disable value	W	Set the auto Tx mute to: (0) Disable or (1) Enabled.	tx_mute_delay	0		√		√	√
./phy_ctrl get bad_data_filter	R	Show whether the bad data filter is: (0) Disable or (1) Enabled.	control			√		√	√
./phy_ctrl set bad_data_filter value	W	Set the bad data filter to: (0) Disable, uncorrectable FEC blocks are allowed to pass thru the DM 256 or (1) Enabled, uncorrectable FEC blocks are discarded by the DM256.	control	1		√		√	√
./phy_ctrl get base_id	R	Shows the current value of the register.	sid			√		√	
./phy_ctrl set base_id value	W	To store the least significant 4 bits of the BS ID received in the DCD message (0-15).	sid			√		√	
./phy_ctrl get base_id_enable	R	Show whether the BS ID filtering is done by the DM 256 is: (0) Disable or (1) Enabled.	sid			√		√	

Read and Write Command Names All Commands for use with ASIC and FPGA Except * ASIC Commands Only	**Read or Write Command**	**Read and Write Command Descriptions**	**Registers Name**	**Settings at Startup**	**Recommended Setting**	**For Customer Use**	**For Wavesat Use**	**Where BS**	**Where SS**
./phy_ctrl set base_id_enable value	W	Set the BS ID filtering by the DM 256 to: (0) Disable or (1) Enabled.	sid	0		√		√	
./phy_ctrl set cfg_symb_err value	W	Set the “Frame Configuration Error” interrupt to: (0) Disable or (1) Enabled.	interrupt	1		√		√	√
./phy_ctrl set config_symb value	W	Set the “FCH Symbol Received” interrupt to: (0) Disable or (1) Enabled.	interrupt	1		√		√	√
./phy_ctrl get cp_size	R	Show the cyclic prefix size at the subscriber station: (64, 32, 16, 8).	cp_size			√			√
./phy_ctrl set cp_size value	W	Set the cyclic prefix size at the subscriber station to: (64, 32, 16, 8).	cp_size	16		√			√
./phy_ctrl get cp_size_enable	R	Not Used. Wavesat use only.				X	X	X	X
./phy_ctrl set data_feedback value	W	Set the data loopback mode to: (0) Disable or (1) Enabled.	control	0			√	√	√
./phy_ctrl get delay_correction	R	Show the number of complex samples set for the Tx delay correction.	delay_conrrection			√			√
./phy_ctrl set delay_correction value	W	Set the Tx delay correction in complex samples (16-bit unsigned value). **Note**: This setting at start up or Wavesat’s recommended settting may have to be changed by customer depending upon the link distance between base and substations.	delay_conrrection	0	1001	√			√
./phy_ctrl get digital_if_force_rx	R	Show whether Rx bandwidth and center frequency are set to: (0) Default values or (1) User-defined values.	digital_if			√			√
./phy_ctrl set digital_if_force_rx value	W	Apply Rx center frequency and bandwidth to: (0) Default or (1) User-defined. **Note**: Default Rx center frequency is ¼ of sampling frequency. Default Rx bandwidth is 3.5 MHz.	digital_if	1		√			√
./phy_ctrl get digital_if_force_tx	R	Show whether Tx bandwidth and center frequency are set to: (0) Default values or (1) User-defined values.	digital_if			√			√

Read and Write Command Names All Commands for use with ASIC and FPGA Except * ASIC Commands Only	**Read or Write Command**	**Read and Write Command Descriptions**	**Registers Name**	**Settings at Startup**	**Recommended Setting**	**For Customer Use**	**For Wavesat Use**	**Where BS**	**Where SS**
./phy_ctrl set digital_if_force_tx value	W	Apply Tx center frequency and bandwidth to: (0) Default or (1) User-defined. **Note**: Default Tx center frequency is ¼ of sampling frequency. Default Tx bandwidth is 3.5 MHz.	digital_if	1		√			√
./phy_ctrl set encode_feedback value	W	Set the encoded word loopback mode to: (0) Disable or (1) Enabled.	control	0			√	√	√
./phy_ctrl get ext_ref	R	Show the reference clock is set to: (0) Internal reference clock or (1) External reference clock.	tx_mute_delay			√		√	√
./phy_ctrl set ext_ref value	W	Set the reference clock to: (0) Internal reference clock or (1) External reference clock.	tx_mute_delay	0		√		√	√
./phy_ctrl get fch_rate_id	R	Shows the Wavesat Rate ID used when decoding FCH bursts.	cp_size			√		√	√
./phy_ctrl set fch_rate_id value	W	Sets the Wavesat Rate ID to be used when decoding FCH bursts. BPSK-1/2 IEEE Rate ID for FCH, Wavesat ID = 6.	cp_size	6		√		√	√
./phy_ctrl set fft_feedback value	W	Not used. Wavesat use only.	control	Not used			X	X	X
./phy_ctrl get force_ref	R	Wavesat use only.	sync_force				√	√	√
./phy_ctrl set force_ref value	W	Wavesat use only.	sync_force	1			√	√	√
./phy_ctrl get force_resync	R	Wavesat use only.	sync_force				√	√	√
./phy_ctrl set force_resync value	W	Wavesat use only.	sync_force	0			√	√	√
./phy_ctrl get force_tx_mute	R	Show whether the Tx mute is: (0) Cancel or (1) Force on.	tx_mute_delay			√		√	√
./phy_ctrl set force_tx_mute value	W	Set the Tx mute to: (0) Cancel or (1) Force on **Note**: Disable the auto Tx mute when using this command.	tx_mute_delay	0		√		√	√
./phy_ctrl get gps_slave	R	Show whether GPS slave is: (0) Disable or (1) Enabled.	control					√	

Read and Write Command Names All Commands for use with ASIC and FPGA Except * ASIC Commands Only	**Read or Write Command**	**Read and Write Command Descriptions**	**Registers Name**	**Settings at Startup**	**Recommended Setting**	**For Customer Use**	**For Wavesat Use**	**Where BS**	**Where SS**
./phy_ctrl set gps_slave value	W	Set the GPS Slave to: (0) Disable or (1) Enabled, allows synchronization of the DL frames to an external GPS 1 pulse per second synchronization signal.	control	0				√	
./phy_ctrl set int_header_err value	W	Set the “Internal Header Error” interrupt to: (0) Disable or (1) Enabled.	interrupt	1		√		√	√
./phy_ctrl get operate_mode	R	Show whether DM 256 is in: (0) Standby/test mode or (1) Normal operating mode.	control			√		√	
./phy_ctrl set operate_mode value	W	Set the DM 256 is in: (0) Standby/test mode or (1) Normal operating mode.	control	1		√		√	
./phy_ctrl set packet_feedback value	W	Set the DM 256 in packet loopback mode: (0) Disable or (1) Enabled.	control	0			√	√	√
./phy_ctrl get pdu_passthrough	R	Show what is being PDU passthrough: (0) All packets or (1) Only data packets.	control			√		√	√
./phy_ctrl set pdu_passthrough value	W	Set the PDU passthrough: (0) All packets or (1) Only data packets.	control	0		√		√	√
./phy_ctrl get ref1_auto_corr_thresh	R	Show the REF1 Auto correlation Threshold.	sync_threshold			√		√	√
./phy_ctrl set ref1_auto_corr_thresh value	W	Set REF1 Auto correlation Threshold.	sync_threshold	125		√		√	√
./phy_ctrl get ref2_auto_corr_thresh	R	Show the REF2 Auto correlation Threshold.	sync_threshold			√		√	√
./phy_ctrl set ref2_auto_corr_thresh value	W	Set REF2 Auto correlation Threshold.	sync_threshold	188		√		√	√
./phy_ctrl get ref2_cross_corr_thresh	R	Show the REF2 Cross correlation Threshold.	sync_threshold			√		√	√
./phy_ctrl set ref2_cross_corr_thresh value	W	Set REF2 Cross correlation Threshold.	sync_threshold	63		√		√	√
./phy_ctrl set reset_irq_stats value	W	To reset the Interrupt statistic counters to: (0) no reset is performed or (1) all interrupt counters are reset =0.	Interrupt			√		√	√
./phy_ctrl get rx_bs_mode	R	Show the Rx link is set in: (0) Subscriber Station mode or (1) Base Station mode.	control			√		√	√

Read and Write Command Names All Commands for use with ASIC and FPGA Except * ASIC Commands Only	**Read or Write Command**	**Read and Write Command Descriptions**	**Registers Name**	**Settings at Startup**	**Recommended Setting**	**For Customer Use**	**For Wavesat Use**	**Where BS**	**Where SS**
./phy_ctrl set rx_bs_mode value	W	Set the Rx channel to: (0) Subscriber Station mode or (1) Base Station mode.	control	1 for BS; 0 for SS		√		√	√
./phy_ctrl set rx_buff_overflow value	W	Set “Rx Buffer Overflow” interrupt to: (0) Disable or (1) Enabled.	interrupt	1		√		√	√
./phy_ctrl get rx_bw_mode	R	Show the Rx channel bandwidth at the subscriber station to: (1.75, 3.5, 7 or 10 MHz).	digital_if			√			√
./phy_ctrl set rx_bw_mode value	W	Set the Rx channel bandwidth at the subscriber station to: (1.75, 3.5, 7 or 10 MHz).	digital_if	3.5		√			√
./phy_ctrl get rx_center_freq	R	Show the Rx center frequency.	digital_if2			√		√	√
./phy_ctrl set rx_center_freq value	W	Set the Rx center frequency.	digital_if2	10000000 (1/4 of sampling clock)		√		√	√
./phy_ctrl get rx_cfg_buff1 value	R	Show the contents of Rx Frame Configuration Buffer 1. **Value** is the length in number of 32 bit words to read.	config_buffers			√		√	√
./phy_ctrl get rx_cfg_buff2 value	R	Show the contents of Rx Frame Configuration Buffer 2. **Value** is the length in number of 32 bit words to read.	config_buffers			√		√	√
./phy_ctrl get rx_buff_1	R	Wavesat use only.	conf_symbol				√	√	√
./phy_ctrl get rx_buff_2	R	Wavesat use only.	conf_symbol				√	√	√
./phy_ctrl get rx_conf_1	R	Wavesat use only.	conf_symbol				√	√	√
./phy_ctrl set rx_conf_1 value	W	Wavesat use only.	conf_symbol				√	√	√
./phy_ctrl get rx_conf_2	R	Wavesat use only.	conf_symbol				√	√	√
./phy_ctrl set rx_conf_2 value	W	Wavesat use only.	conf_symbol				√	√	√
./phy_ctrl get rx_adc_mode *	R	Show the Rx ADC digital bus mode: (1) Enable 2’s complement or (0) Enable straight bit format.	digital_if			√		√	√
./phy_ctrl set rx_adc_mode value *	W	To set the Rx ADC digital bus mode: (1) Enable 2’s complement (default for built in ADCs) or (0) Enable straight bit format.	digital_if	0		√		√	√
./phy_ctrl get rx_data_fifo_sync_reset	R	Show whether the Rx payload buffer is: (1) Enable, which causes a Reset of the buffer or (0) Normal setting, which does not cause a clear of the buffer.	control			√		√	√

Read and Write Command Names All Commands for use with ASIC and FPGA Except * ASIC Commands Only	**Read or Write Command**	**Read and Write Command Descriptions**	**Registers Name**	**Settings at Startup**	**Recommended Setting**	**For Customer Use**	**For Wavesat Use**	**Where BS**	**Where SS**
./phy_ctrl set rx_data_fifo_sync_reset value	W	To reset the Rx payload buffer: (1) Enable reset or (0) Normal setting, not cleared.	control	0		√		√	√
./phy_ctrl get rx_data_ready	R	Show the state of the Rx payload buffer: (0) The buffer is < ¾ full and is not ready for read or (1) The buffer is ≥ ¾ full or more, and is ready for read.	interrupt			√		√	√
./phy_ctrl set rx_data_ready value	W	Set Rx Data Buffer Ready interrupt to: (0) Disable or (1) Enabled.	interrupt	0		√		√	√
./phy_ctrl get rx_deinterleaver	R	Show whether the Rx deinterleaver is: (0) Disable or (1) Enabled.	control			√		√	√
./phy_ctrl set rx_deinterleaver value	W	Set Rx deinterleaver to: (0) Disable or (1) Enabled.	control	0 (enabled)		√		√	√
./phy_ctrl get rx_derandomizer	R	Show whether the Rx derandomizer is: (0) Disable or (1) Enabled.	control			√		√	√
./phy_ctrl set rx_derandomizer value	W	Set Rx derandomizer to: (0) Disable or (1) Enabled.	control	0 (enabled)		√		√	√
./phy_ctrl get rx_digital_iq_output *	R	Show whether the Rx Frequency input is: (0) I/Q input mode or (1) IF input mode. **Note:** Error in command syntax: ./phy_ctrl get rx_digital_iq_output output vs. input. In future releases of software this will be corrected to ./phy_ctrl get rx_digital_iq_output.	digital_if						
./phy_ctrl set rx_digital_iq_output value *	W	Set the Rx Frequency input to: (0) I/Q input mode or (1) IF input mode. **Note:** Error in command syntax: ./phy_ctrl set rx_digital_iq_output value output vs. input. In future releases of software this will be corrected to ./phy_ctrl set rx_digital_iq_output value.	digital_if	0					

Read and Write Command Names All Commands for use with ASIC and FPGA Except * ASIC Commands Only	Read or Write Command	Read and Write Command Descriptions	Registers Name	Settings at Startup	Recommended Setting	For Customer Use	For Wavesat Use	Where BS	Where SS
./phy_ctrl get rx_dma	R	Show DMA request status of the Rx payload is: (0) Disable or (1) Enabled.	dma1			√		√	√
./phy_ctrl set rx_dma value	W	Set the Rx payload DMA request to: (0) Disable, CPU request/transfer data to DM256 without DM but using the 32 bits parallel bus or (1) Enabled, CPU request/transfer data to DM 256 using the CPU DMA Engine, which is more efficient.	dma1	1		√		√	√
./phy_ctrl get rx_dma_transf_width	R	Show the DMA transfer width of the Rx Payload: (0) word, (1) byte, (2) half-word, or (3) word	dma2			√		√	√
./phy_ctrl set rx_dma_transf_width value	W	Set the Rx payload DMA transfer width to: (0) word, (1) byte, (2) half-word, or (3) word	dma2	1		√		√	√
./phy_ctrl get rx_fifo	R	Read the contents (next 32 bits) of the Rx payload buffer.	rx_fifo			√		√	√
./phy_ctrl get rx_fifo_count	R	Show the content size of the Rx payload buffer.	dma2			√		√	√
./phy_ctrl get rx_force_gain	R	Not used. Wavesat use only.	config	Not used			X	X	X
./phy_ctrl set rx_force_gain value	W	Not used. Wavesat use only.	config	Not used			X	X	X
./phy_ctrl get rx_force_gain_mode	R	Not used. Wavesat use only.	config	Not used			X	X	X
./phy_ctrl set rx_force_gain_mode value	W	Not used. Wavesat use only.	config	Not used			X	X	X
./phy_ctrl set rx_frame value	W	Set the End of Rx Frame interrupt: (0) Disable or (1) Enabled.	interrupt	1		√		√	√
./phy_ctrl get rx_freq_corrector	R	Wavesat use only.	control				√	√	√
./phy_ctrl set rx_freq_corrector value	W	Wavesat use only.	control	0			√	√	√
./phy_ctrl get rx_freq_ratio	R	Show the Rx frequency ratio.	digital_if2	TBD		√		√	√
./phy_ctrl set rx_freq_ratio value	W	Set the Rx frequency ratio.	digital_if2	TBD		√		√	√
./phy_ctrl get rx_gain_value	R	Not used. Wavesat use only.	config_value	Not used			X	X	X
./phy_ctrl set rx_gain_value value	W	Not used. Wavesat use only.	config_value	Not used			X	X	X
./phy_ctrl get rx_inner_decoder	R	Not used. Wavesat use only.	control	Not used			X	X	X

<table>
<tr><th>Read and Write Command Names
All Commands for use with ASIC and FPGA Except * ASIC Commands Only</th><th>Read or Write Command</th><th>Read and Write Command Descriptions</th><th>Registers Name</th><th>Settings at Startup</th><th>Recommended Setting</th><th>For Customer Use</th><th>For Wavesat Use</th><th>Where BS</th><th>Where SS</th></tr>
<tr><td>./phy_ctrl set rx_inner_decoder value</td><td>W</td><td>Not used. Wavesat use only.</td><td>control</td><td>Not used</td><td></td><td></td><td>X</td><td>X</td><td>X</td></tr>
<tr><td>./phy_ctrl get rx_link_active</td><td>R</td><td>Show whether the Rx link is currently:
(0) Inactive or
(1) Active.</td><td>status</td><td></td><td></td><td>√</td><td></td><td>√</td><td>√</td></tr>
<tr><td>./phy_ctrl get rx_link_config</td><td>R</td><td>Show whether Rx Link is currently:
(0) Not Configured or
(1) Configured.</td><td>status</td><td></td><td></td><td>√</td><td></td><td>√</td><td>√</td></tr>
<tr><td>./phy_ctrl get rx_noise_inducer</td><td>R</td><td>Show the state of the Rx Noise Generator:
(0) Disabled or
(1) Enabled.</td><td>control</td><td></td><td></td><td>√</td><td></td><td>√</td><td>√</td></tr>
<tr><td>./phy_ctrl set rx_noise_inducer value</td><td>W</td><td>Set the state of the Rx Noise Generator to:
(0) Disabled or
(1) Enabled.
Note: The noise generator should be set on when in sample feedback mode.</td><td>control</td><td>1</td><td>0</td><td>√</td><td></td><td>√</td><td>√</td></tr>
<tr><td>./phy_ctrl get rx_outer_decoder</td><td>R</td><td>Not used. Wavesat use only.</td><td>control</td><td>Not used</td><td></td><td></td><td>X</td><td>X</td><td>X</td></tr>
<tr><td>./phy_ctrl set rx_outer_decoder value</td><td>W</td><td>Not used. Wavesat use only.</td><td>control</td><td>Not used</td><td></td><td></td><td>X</td><td>X</td><td>X</td></tr>
<tr><td>./phy_ctrl get rx_phase_corrector</td><td>R</td><td>Wavesat use only.</td><td>control</td><td>0</td><td></td><td></td><td>√</td><td>√</td><td>√</td></tr>
<tr><td>./phy_ctrl set rx_phase_corrector value</td><td>W</td><td>Wavesat use only.</td><td>control</td><td>0</td><td></td><td></td><td>√</td><td>√</td><td>√</td></tr>
<tr><td>./phy_ctrl get rx_scaling</td><td>R</td><td>Show the Rx scaling (4-bit 2’s complement unsigned value). Ranging from digital -8 to +7.
<table><tr><td>Un signed</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr><tr><td>Signed</td><td>-1</td><td>-2</td><td>-3</td><td>-4</td><td>-5</td><td>-6</td><td>-7</td><td>-8</td></tr><tr><td>Un signed</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr><tr><td>Signed</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr></table></td><td>papr</td><td></td><td></td><td>√</td><td></td><td>√</td><td>√</td></tr>
<tr><td>./phy_ctrl set rx_scaling value</td><td>W</td><td>Set the Rx scaling (4-bit 2’s complement signed value).</td><td>papr</td><td>-1</td><td></td><td>√</td><td></td><td>√</td><td>√</td></tr>
<tr><td>./phy_ctrl set sample_feedback value</td><td>W</td><td>Set the sample loopback mode to:
(0) Disabled or
(1) Enabled.</td><td>control</td><td>0</td><td></td><td></td><td>√</td><td>√</td><td>√</td></tr>
</table>

Read and Write Command Names All Commands for use with ASIC and FPGA Except * ASIC Commands Only	**Read or Write Command**	**Read and Write Command Descriptions**	**Registers Name**	**Settings at Startup**	**Recommended Setting**	**For Customer Use**	**For Wavesat Use**	**Where BS**	**Where SS**
./phy_ctrl get serial_mode	R	Show the status of the serial port mode: (0) Disabled or (1) Enabled.	control			√		√	√
./phy_ctrl set serial_mode value	W	Set the status of the serial port mode to: (0) Disabled or (1) Enabled.	control	0		√		√	√
./phy_ctrl get serial_port	R	Show the status of the serial port: (0) Disabled or (1) Enabled.	control			√		√	√
./phy_ctrl set serial_port value	W	Set the status of the serial port to: (0) Disabled or (1) Enabled.	control	0		√		√	√
./phy_ctrl get snr_power	R	To return the Signal to Noise Ratio and Power level on the symbol immediately following a REF2 symbol type. **Note** that this command returns 4X the actual value.	snr_power			√		√	√
./phy_ctrl get spi_clock_enable	R	Show whether the SPI clock is set to: (0) Default value = 6.250 MHz or (1) User Defined.	spi			√		√	√
./phy_ctrl set spi_clock_enable value	W	Set the SPI clock to: (0) Default value = 6.250 MHz or (1) User Defined.	spi	0		√		√	√
./phy_ctrl get spi_clock_speed	R	Show the setting of the SPI Clock.	spi			√		√	√
./phy_ctrl set spi_clock_speed value	W	Set the SPI clock: 0 = 6.250 MHz 1 = 3.125 MHz 2 = 1.562 MHz 3 = 781.250 kHz 4 = 390.625 kHz 5 = 195.312 kHz 6 = 97.656 kHz 7 = 48.828 kHz 8 = 24.414 kHz 9 = 12.207 kHz 10 –15 = 6.250 MHz	spi	0		√		√	√
./phy_ctrl get spi_enable	R	Show whether the SPI interface is: (0) Disabled or (1) Enabled.	spi			√		√	√
./phy_ctrl set spi_enable value	W	Set the SPI interface to: (0) Disabled or (1) Enabled.	spi	1		√		√	√
./phy_ctrl get spi_manual_mode_enable	R	Show whether the SPI is set to: (0) Automatic Mode or (1) Manual Mode.	spi			√		√	√

Read and Write Command Names All Commands for use with ASIC and FPGA Except * ASIC Commands Only	**Read or Write Command**	**Read and Write Command Descriptions**	**Registers Name**	**Settings at Startup**	**Recommended Setting**	**For Customer Use**	**For Wavesat Use**	**Where BS**	**Where SS**
./phy_ctrl set spi_manual_mode_enable value	W	Set the SPI to: (0) Automatic Mode or (1) Manual Mode.	spi	1		√		√	√
./phy_ctrl set spi_pll_dac value	W	Set the PLL DAC (0 – 255).	spi	128		√		√	√
./phy_ctrl set spi_rx_dac value	W	Set the Rx DAC (0 – 255).	spi	105		√		√	√
./phy_ctrl set spi_tx_dac value	W	Set the Tx DAC (0 – 255).	spi	90		√		√	√
./phy_ctrl get test_port	R	Wavesat use only.	control				√	√	√
./phy_ctrl set test_port value	W	Wavesat use only.	control	0			√	√	√
./phy_ctrl get tx_bs_mode	R	Show whether the Tx link is set to: (0) Subscriber Station mode or (1) Base Station mode.	control			√		√	√
./phy_ctrl set tx_bs_mode value	W	Set the Tx channel to: (0) Subscriber Station mode or (1) Base Station mode.	control	1 for BS; 0 for SS		√		√	√
./phy_ctrl get tx_bw_mode	R	Show the Tx channel bandwidth which is set at the subscriber station. (1.75, 3.5, 7 or 10 MHz)	digital_if			√			√
./phy_ctrl set tx_bw_mode value	W	Set the Tx channel bandwidth at the subscriber station (1.75, 3.5, 7 or 10 MHz).	digital_if	3.5		√			√
./phy_ctrl get tx_center_freq	R	Show the Tx center frequency.	digital_if2			√		√	√
./phy_ctrl set tx_center_freq value	W	Set the Tx center frequency.	digital_if2	10000000 (1/4 of sampling clock)		√		√	√
./phy_ctrl get tx_cfg_buff1 value	R	Shows the content of Tx configuration buffer 1. **Value** is the length in number of words (32 bits) to read.	config_buffers			√		√	√
./phy_ctrl get tx_cfg_buff2 value	R	Shows the content of Tx configuration buffer 2. **Value** is the length in number of words (32 bits) to read.	config_buffers			√		√	√
./phy_ctrl get tx_conf_1	R	Wavesat use only.	conf_symbol				√	√	√
./phy_ctrl set tx_conf_1 value	W	Wavesat use only.	conf_symbol	0			√	√	√
./phy_ctrl get tx_conf_2	R	Wavesat use only.	conf_symbol				√	√	√
./phy_ctrl set tx_conf_2 value	W	Wavesat use only.	conf_symbol	0			√	√	√

Read and Write Command Names All Commands for use with ASIC and FPGA Except * ASIC Commands Only	**Read or Write Command**	**Read and Write Command Descriptions**	**Registers Name**	**Settings at Startup**	**Recommended Setting**	**For Customer Use**	**For Wavesat Use**	**Where BS**	**Where SS**
./phy_ctrl get tx_cont_wave	R	Show whether center frequency at the subscriber station is: (0) in test mode for analog side, sin wave at Center Frequency or (1) normal operation mode.	digital_if				√		√
./phy_ctrl set tx_cont_wave value	W	To create a dc carrier at the set center frequency at the subscriber station: (0) in test mode for analog side, sin wave at Center Frequency set according to./phy_ctrl set tx_freq_ratio command or (1) normal operation mode.	digital_if	0			√		√
./phy_ctrl get tx_dac_mode *	R	Show whether Tx DAC digital bus mode is set to: (0) Enable straight bit format or (1) Enable 2’s complement (default for built in DACs).	digital_if			√			√
./phy_ctrl set tx_dac_mode value *	W	Set the Tx DAC digital bus mode to: (0) Enable straight bit format or (1) Enable 2’s complement (default for built in DACs).	digital_if	0	1	√			√
./phy_ctrl get tx_data_fifo_sync_reset	R	Show the Rx data payload buffer is: (0) Not Cleard or (1) Cleared.	control			√		√	√
./phy_ctrl set tx_data_fifo_sync_reset value	W	Reset the Tx payload buffer to: (0) Normal setting or (1) Enable.	control	0		√		√	√
./phy_ctrl get tx_data_ready	R	Show the state of the Tx payload buffer: (0) The buffer is more than 1/4 full and is not ready for write or (1) The buffer is 1/4 full or less, and is ready for write.	interrupt			√		√	√
./phy_ctrl set tx_data_ready value	W	To set the Tx Data Buffer Ready” interrupt: (0) Disabled or (1) Enabled.	interrupt	1		√		√	√
./phy_ctrl get tx_digital_iq_output	R	Show the Tx Frequency Output.	digital_if			√		√	√
./phy_ctrl set tx_digital_iq_output value	W	To apply Tx Frequency Output: (0) IF or (1) I/Q.	digital_if	0		√		√	√
./phy_ctrl get tx_dma	R	Show the DMA request status of the Tx Payload: (0) Disabled or (1) Enabled.	dma1			√		√	√

Read and Write Command Names All Commands for use with ASIC and FPGA Except * ASIC Commands Only	**Read or Write Command**	**Read and Write Command Descriptions**	**Registers Name**	**Settings at Startup**	**Recommended Setting**	**For Customer Use**	**For Wavesat Use**	**Where BS**	**Where SS**
./phy_ctrl set tx_dma value	W	To set the “Tx Payload Request” interrupt: (0) Disabled or (1) Enabled.	dma1	1		√		√	√
./phy_ctrl get tx_enable	R	Show whether Tx Enable mode is: (0) Disabled or (1) Enabled.	control			√		√	√
./phy_ctrl set tx_enable value	W	To set Tx Enable mode to: (0) Disabled or (1) Enabled.	control	1		√		√	√
./phy_ctrl set tx_fifo value	W	Write 32-bit of decimal data to the Tx payload buffer.	fifo				√	√	√
./phy_ctrl set tx_fifo_reset value	W	To set the “Tx Payload Buffer Reset” interrupt: (0) Disabled or (1) Enabled.	interrupt	1		√		√	√
./phy_ctrl set tx_fifo_underrun value	W	Enables generation of an interrupt when the DM 256 is scheduled to generate an OFDM symbol for Tx and there is insufficient space in the Tx FIFO to perform the operation. Symbol will be padded with OxFF. (0) Disabled or (1) Enabled.	interrupt			√		√	√
./phy_ctrl get tx_fifo_count	R	Show the content size of the Tx payload buffer.	dma2			√		√	√
./phy_ctrl get tx_force_gain	R	Show whether the Tx digital gain factor is set to: (0) Use default value (255) or (1) User-defined digital gain.	config			√		√	√
./phy_ctrl set tx_force_gain value	W	Set the Tx digital gain factor is set to: (0) Use default value (255) or (1) User-defined digital gain.	config	1		√		√	√
./phy_ctrl set tx_frame value	W	To set the “End of Tx Frame” interrupt: (0) Disabled or (1) Enabled.	interrupt	1		√		√	√
./phy_ctrl get tx_freq_ratio	R	Show the Tx frequency ratio.	digital_if2			√		√	√
./phy_ctrl set tx_freq_ratio value	W	Set the Tx frequency ratio.	digital_if2	TBD		√		√	√
./phy_ctrl get tx_gain_value	R	Show the Tx digital gain.	config_value			√		√	√
./phy_ctrl set tx_gain_value value	W	Set the Tx digital gain (0 – 255).	config_value	250		√		√	√
./phy_ctrl get tx_inner_encoder	R	Not used. Wavesat use only.	control	Not used			X	X	X
./phy_ctrl set tx_inner_encoder value	W	Not used. Wavesat use only.	control	0			X	X	X

<table>
<tr><th>Read and Write Command Names
All Commands for use with ASIC and FPGA
Except * ASIC Commands Only</th><th>Read or Write Command</th><th>Read and Write Command Descriptions</th><th>Registers Name</th><th>Settings at Startup</th><th>Recommended Setting</th><th>For Customer Use</th><th>For Wavesat Use</th><th>Where BS</th><th>Where SS</th></tr>
<tr><td>./phy_ctrl get tx_interleaver</td><td>R</td><td>Show whether the Tx interleaver is:
(0) Disabled or
(1) Enabled.</td><td>control</td><td></td><td></td><td>√</td><td></td><td>√</td><td>√</td></tr>
<tr><td>./phy_ctrl set tx_interleaver value</td><td>W</td><td>Set the Tx interleaver to:
(0) Disabled or
(1) Enabled.</td><td>control</td><td>0 (enabled)</td><td></td><td>√</td><td></td><td>√</td><td>√</td></tr>
<tr><td>./phy_ctrl get tx_mute_delay_lag</td><td>R</td><td>Show the Tx mute delay lag.</td><td>tx_mute_delay</td><td></td><td></td><td>√</td><td></td><td>√</td><td>√</td></tr>
<tr><td>./phy_ctrl set tx_mute_delay_lag value</td><td>W</td><td>Set the Tx mute delay lag (0 – 255).</td><td>tx_mute_delay</td><td>4</td><td></td><td>√</td><td></td><td>√</td><td>√</td></tr>
<tr><td>./phy_ctrl get tx_mute_delay_lead</td><td>R</td><td>Show the Tx mute delay lead.</td><td>tx_mute_delay</td><td></td><td></td><td>√</td><td></td><td>√</td><td>√</td></tr>
<tr><td>./phy_ctrl set tx_mute_delay_lead value</td><td>W</td><td>Set the Tx mute delay lead (0 – 255).</td><td>tx_mute_delay</td><td>60</td><td></td><td>√</td><td></td><td>√</td><td>√</td></tr>
<tr><td>./phy_ctrl get tx_outer_encoder</td><td>R</td><td>Not used. Wavesat use only.</td><td>control</td><td>Not used</td><td></td><td></td><td>X</td><td>X</td><td>X</td></tr>
<tr><td>./phy_ctrl set tx_outer_encoder value</td><td>W</td><td>Not used. Wavesat use only.</td><td>control</td><td>0</td><td></td><td></td><td>X</td><td>X</td><td>X</td></tr>
<tr><td>./phy_ctrl get tx_rand_stream</td><td>R</td><td>To show whether the PRBS 23 is:
(0) Cancelled or
(1) Generated.</td><td>control</td><td></td><td></td><td>√</td><td></td><td>√</td><td>√</td></tr>
<tr><td>./phy_ctrl set tx_rand_stream value</td><td>W</td><td>To set the PRBS 23 data generation for BER test to:
(0) Cancelled or
(1) Generated..</td><td>control</td><td>0</td><td></td><td>√</td><td></td><td>√</td><td>√</td></tr>
<tr><td>./phy_ctrl get tx_randomizer</td><td>R</td><td>Show whether the Tx randomizer is:
(0) Disabled or
(1) Enabled.</td><td>control</td><td></td><td></td><td>√</td><td></td><td>√</td><td>√</td></tr>
<tr><td>./phy_ctrl set tx_randomizer value</td><td>W</td><td>To set the Tx randomizer to:
(0) Disabled or
(1) Enabled.</td><td>control</td><td>0 (enabled)</td><td></td><td>√</td><td></td><td>√</td><td>√</td></tr>
<tr><td>./phy_ctrl get tx_scaling</td><td>R</td><td>Show the Tx scaling (4-bit 2's complement signed value)
<table>
<tr><td>Un signed</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td>Signed</td><td>-1</td><td>-2</td><td>-3</td><td>-4</td><td>-5</td><td>-6</td><td>-7</td><td>-8</td></tr>
<tr><td>Un signed</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>Signed</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
</table></td><td>papr</td><td></td><td></td><td>√</td><td></td><td>√</td><td>√</td></tr>
<tr><td>./phy_ctrl set tx_scaling value</td><td>W</td><td>Set the Tx scaling (4-bit 2's complement signed value).</td><td>papr</td><td>-1</td><td></td><td>√</td><td></td><td>√</td><td>√</td></tr>
</table>

Read and Write Command Names All Commands for use with ASIC and FPGA Except * ASIC Commands Only	Read or Write Command	Read and Write Command Descriptions	Registers Name	Settings at Startup	Recommended Setting	For Customer Use	For Wavesat Use	Where BS	Where SS
./phy_ctrl get wavesat_cfg_sym_mode	R	Not used. Wavesat use only.				X	X	X	X
./phy_ctrl set wavesat_cfg_sym_mode value	W	Not used. Wavesat use only.				X	X	X	X
./phy_ctrl version	R	Show the version of the DM256.	dac			√		√	√
./phy_ctrl get vga	R	Show value for VGA algorithm in the AGC module. The returned value is the inverse of the actual value.	dac			√			√
./phy_ctrl set vga value	W	Set a constant value for VGA algorithm in the AGC module. Range of 20-90 db, for Wavesat's receive amplifier VGA gain 50 db = value 81.	dac	81		√			√
./sched_ctrl get bert	R	Show whether mode of operation for testing with PRBS Analyzer is: (0) Disable or (1) Enable.				√		√	
./sched_ctrl set bert value	W	Set the mode of operation for testing with PRBS Analyzer to: (0) Disable or (1) Enable (when using PRBS Analyzer).		0		√		√	
./sched_ctrl get channel_bw	R	Show the channel bandwidth set at the base station.				√		√	
./sched_ctrl set channel_bw value	W	Set the channel bandwidth at the base station to: (1.75, 3.5, 7 or 10)		3.5		√		√	
./sched_ctrl get contention_slot	R	Show a contention slot is: (0) Not added to the Uplink or (1) Added to the Uplink.				√		√	
./sched_ctrl set contention_slot value	W	Add a contention slot to: (0) Do Not Add or (1) Uplink.		1		√		√	
./sched_ctrl get cp_size	R	Show the cyclic prefix size at the base station. (1/4, 1/8, 1/16 or 1/32).				√		√	
./sched_ctrl set cp_size value	W	Set the cyclic prefix size at the base station to (1/4, 1/8, 1/16 or 1/32).		1/16		√		√	
./sched_ctrl get dcd_change_count	R	Wavesat Use Only.					√	√	
./sched_ctrl set dcd_change_count value	W	Wavesat Use Only.					√	√	
./sched_ctrl get debug	R	Wavesat Use Only.					√	√	
./sched_ctrl set debug value	W	Wavesat Use Only.		0			√	√	
./sched_ctrl get default_rate_id	R	Wavesat Use Only.					√	√	

Read and Write Command Names All Commands for use with ASIC and FPGA Except * ASIC Commands Only	**Read or Write Command**	**Read and Write Command Descriptions**	**Registers Name**	**Settings at Startup**	**Recommended Setting**	**For Customer Use**	**For Wavesat Use**	**Where BS**	**Where SS**
./sched_ctrl get duplex_mode	R	Show duplex mode is set to: (tdd) TDD or (fdd) FDD.				√		√	
./sched_ctrl set duplex_mode mode	W	Set duplex mode to: (tdd) TDD or (fdd) FDD.		tdd		√		√	
./sched_ctrl get FEC	R	Show whether the Forward Error Correction (FEC) to: (0) Disable or (1) Enable.				√		√	
./sched_ctrl set FEC value	W	Set the Forward Error Correction (FEC) to: (0) Disable or (1) Enable.		1		√		√	
./sched_ctrl get feedback_mode	R	Show whether the scheduler and the DM 256 are in a specific feedback mode: (none), normal operating mode or (data, packet, encoder, sample, if), in test mode only.				√		√	
./sched_ctrl set feedback_mode value	W	Set the scheduler and the DM 256 into a specific feedback mode: (none), normal operating mode. or (data, packet, encoder, sample, if), in test mode only.		none		√		√	
./sched_ctrl get frame_code	R	Show the frame code.				√		√	
./sched_ctrl set frame_code value	W	Set the frame code (0,1,2,3,4,5,6).		4		√		√	
./sched_ctrl get gap	R	Wavesat Use Only.					√	√	
./sched_ctrl set gap value	W	Wavesat Use Only.					√	√	
./sched_ctrl get operating_mode	R	Wavesat Use Only.					√	√	
./sched_ctrl set operating_mode value	W	Wavesat Use Only.					√	√	
./sched_ctrl get pause	R	Show whether Data transmission pause is: (0) Deactivate or (1) Activate.				√		√	
./sched_ctrl set pause value	W	Set the Data transmission to: (0) Deactivate the pause or (1) Activate the pause.		0		√		√	
./sched_ctrl get rtg	R	Show the RTG (Receive to Transmit GAP) (TDD only).				√		√	
./sched_ctrl set rtg value	W	Set the RTG (0 – 255) (Receive to Transmit GAP) (TDD only).		20		√		√	

Read and Write Command Names All Commands for use with ASIC and FPGA Except * ASIC Commands Only	**Read or Write Command**	**Read and Write Command Descriptions**	**Registers Name**	**Settings at Startup**	**Recommended Setting**	**For Customer Use**	**For Wavesat Use**	**Where BS**	**Where SS**
./sched_ctrl get status	R	Wavesat Use Only.					√	√	
./sched_ctrl set start value	W	Start the data transmission and set the transmission delay between frames.		0		√		√	
./sched_ctrl get test_stc_zone	R	Wavesat Use Only.					√	√	
./sched_ctrl set test_stc_zone value	W	Wavesat Use Only.					√	√	
./sched_ctrl get ttg	R	Show the TTG (Transmit to Receive GAP) (TDD only).				√		√	
./sched_ctrl set ttg value	W	Set the TTG (0 – 255) (Transmit to Receive GAP) (TDD only).		20		√		√	
./sched_ctrl print_get	R	Show all commands with ./sched_ctrl get.				√		√	
./sched_ctrl print_set	R	Show all commands with ./sched_ctrl set.				√		√	
./sched_ctrl print_user	R	Show all commands with ./sched_ctrl user.				√		√	
./sched_ctrl user add [CID] [DL_Modulation] [#DL_Allocation] [UL_Modulation] [#UL_Allocation] Preamble Midamble	W	Wavesat Use Only. Do Not Use.				X	√	X	
./sched_ctrl user del [CID] [DL_Modulation] [#DL_Allocation] [UL_Modulation] [#UL_Allocation] Preamble Midamble	W	Wavesat Use Only. Do Not Use.				X	√	X	
./sched_ctrl user modify [CID] [DL_Modulation] [#DL_Allocation] [UL_Modulation] [#UL_Allocation] Preamble Midamble	W	Wavesat Use Only. *** Use only for BER Testing. See 10.7 BER Testing for more details.				√ ***	√	X	
./sched_ctrl user print [CID] [DL_Modulation] [#DL_Allocation] [UL_Modulation] [#UL_Allocation] Preamble Midamble	W	Wavesat Use Only. Do Not Use.				X	√	X	
./sched_ctrl version	R	Show the version of the scheduler.				√		√	
./ss_linktest_ctrl add CID type	W	To add a Connection ID for received software –based PRBS data. Also used by map processor to filter out unwanted connections. Add a connection ID (1 – 36). Type: Transport, LInkTest or Broadcast.	linktest			√			√
./ss_linktest_ctrl del CID	W	Delete a connection ID (1 – 36).				√			√

Read and Write Command Names All Commands for use with ASIC and FPGA Except * ASIC Commands Only	Read or Write Command	Read and Write Command Descriptions	Registers Name	Settings at Startup	Recommended Setting	For Customer Use	For Wavesat Use	Where BS	Where SS
./ss_linktest_ctrl show	R	Show the Connection ID and the Connection Type (Transport, LinkTest or Broadcast) used by the Subscriber Station.				√			√

4.5 Scheduling Parameters

The following parameters affect the way the scheduling is done and thus cannot be modified once the scheduling has started; i.e. the "./sched_ctrl set start" command is executed (set to 0).

Table 7 - Parameters Which Affect Scheduling

Parameters affected after "./sched_ctrl set start" command is executed (set to 0).
Channel bandwidth
CP size
Duplex mode
FEC
Feedback
Frame code
RTG
TTG

Therefore if you wish to modify any of these above parameters, you need to:

1. Reboot the development kit.
2. Change the parameter settings in the base station script.
3. Apply the script.

Note: In the script, the "./sched_ctrl set start" command must appear last.

Table 8 - Parameters and Commands Which Affect Scheduling

Parameter	Set via the corresponding command
Channel bandwidth	./sched_ctrl set channel_bw
CP size	./sched_ctrl set cp_size
Duplex mode	./sched_ctrl set duplex_mode
FEC	./sched_ctrl set FEC
Feedback	./sched_ctrl set feedback_mode
Frame code	./sched_ctrl set frame_code
RTG	./sched_ctrl set rtg
TTG	./sched_ctrl set ttg

This page has been left intentionally blank.

Section: 5 Custom Startup

Note:	Most users do not need this section.

This section provides the necessary information for users who wish to perform the following:

- Modify the bootstrap.
- Upgrade the flash root file system (JFFS2).
- Upgrade the kernel.
- Upgrade the FPGA file.
- Upgrade the command interface software.
- Programming the FPGA.
- Modify the base station or subscriber station script.

5.1 Modify the Bootstrap

To modify the bootstrap of each development kit, proceed as follows:

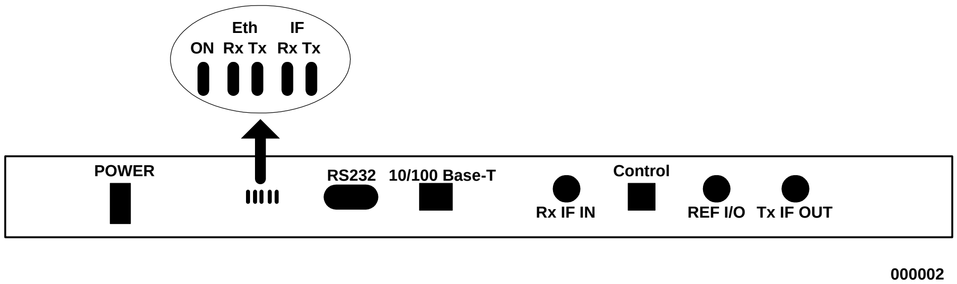


Figure 7 - Console Interface

1. Turn the development kit On by toggling the POWER switch on the faceplate of the development kit.

   The ON LED will illuminate.

2. Before or while the message below appears on the screen, press any key to stop the booting process.

   "Hit any key to stop autoboot"

-Continue-

-

3. To display the bootstrap settings, type:

```
print ↵
```

Note:	The Enter key is noted by the following symbol: ↵

The bootstrap settings such as the following appear:

```
=>print <enter>
bootdelay=3
baudrate=9600
loads_echo=1
ethaddr=00:00:00:06:00:cc
ofdmaddr=00:00:00:06:01:cc
ipaddr=10.1.10.150
serverip=10.1.10.129
bootcmd=fsload /boot/pImage ; bootm 0x100000
bootargs=root=/dev/mtdblock0 rw ip=10.1.10.150:10.1.10.129:255.255.0.0
console=ttyS0,9600
stdin=serial
stdout=serial
stderr=serial
```

4. To change the Ethernet address, type:

```
setenv ethaddr 00:00:00:xx:yy:zz ↵
```

Where

Parameter	Description
xx	is the board revision
yy	00 for Ethernet and 01 for OFDM
zz	hexadecimal value of the serial number of the board

5. To change the OFDM address, type:

```
setenv ofdmaddr 00:00:00:xx:yy:zz ↵
```

Where

Parameter	Description
xx	is the board revision
yy	00 for Ethernet and 01 for OFDM
zz	hexadecimal value of the serial number of the board

-Continue-

6. To change the server IP address, type:

```
setenv serverip aa.bb.cc.dd ↵
```

Where

Parameter	Description
aa.bb.cc.dd	is the IP address of the server used for file transfer

7. To change the development kit IP address, type:

```
setenv ipaddr aa.bb.cc.dd ↵
```

Where

Parameter	Description
aa.bb.cc.dd	is the IP address of the development kit

8. To change the boot arguments used when booting the development kit, type:

```
setenv bootargs root=/dev/mtdblock0 rw
ip=aa.bb.cc.dd:ee.ff.gg.hh:255.255.0.0 console=ttyS0,9600 ↵
```

Note:	Ensure that you have typed in the entire command until the symbol ↵ where you would press the enter key.

Where

Parameter	Description
aa.bb.cc.dd	is the development kit IP address
ee.ff.gg.hh	is the server IP address
255.255.0.0	is the netmask

9. If you have made any modifications, save them by typing:

```
saveenv ↵
```

The following message appears:

Saving Environment to EEPROM

=>

10. Reset the development kit by restarting it or by typing the following command:

```
reset ↵
```

-End-

5.2 Upgrading the Journaling Flash Root File System (JFFS2)

To upgrade the Journaling Flash File System 2 (JFFS2) of each development kit, proceed as follows:

1. Ensure that both the development kit and the computer from which you wish to download the JFFS2.img file are connected to the local network. The 10/100 Base T port is used to connect the development kit to the local network.

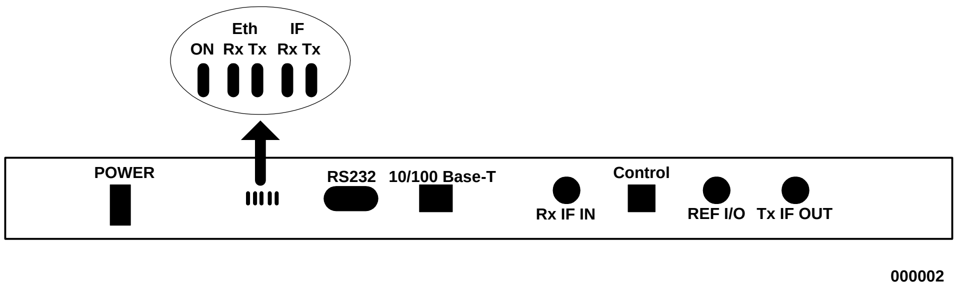


Figure 8 - Console Interface

2. Configure your computer as a TFTP server.
3. Download the JFFS2.img file from the Wavesat website.
4. Save the JFFS2.img file in the TFTP server download directory.

-Continue-

5. Ensure that the TFTP server is properly configured.

   For example: ensure that it is configured for both transmitting and receiving files and that it will accept the IP address of the development kit.

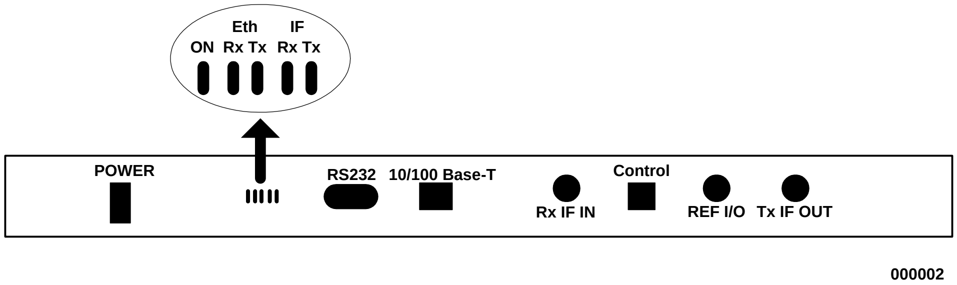


Figure 9 - Console Interface

6. Turn the development kit On by toggling the POWER switch on the faceplate of the development kit.

   The ON LED will illuminate.

7. Before or while the message below appears on the screen, press any key to stop the booting process.

   "Hit any key to stop autoboot"

8. Verify that the Serverip = local TFTP server IP Address.

9. Before you program the new JFFS2 image file into the flash memory, unprotect the flash by typing the following command:

```
protect off all ↵
```

10. Erase the content of the flash by typing the following command:

```
erase all ↵
```

Note:	This command will erase the contents of the flash which includes the Linux flash file syste with its files.

11. Perform a TFTP operation to download the JFFS2 image file to flash memory by typing the following command:

```
tftp 0x8010000 jffs2.img ↵
```

-Continue-

12. Configure the bootargs parameters by typing the following command:

```
setenv bootargs root=/dev/mtdblock0 rw
ip=aa.bb.cc.dd:ee.ff.gg.hh:255.255.0.0 console=ttyS0,9600 ↵
```

Note:	Ensure that you have typed in the entire command until the symbol ↵ where you would press the enter key.

Where

Parameter	Description
aa.bb.cc.dd	is the development kit IP address
ee.ff.gg.hh	is the server IP address
255.255.0.0	is the netmask

13. Configure the bootcmd parameter by typing the following command:

```
setenv bootcmd fsload /boot/pImage \; bootm 0x100000 ↵
```

14. Save the modifications by typing the following command:

```
saveenv ↵
```

The following message appears:

Saving Environment to EEPROM

=>

15. Reset the development kit by restarting it or by typing the following command:

```
reset ↵
```

-End-

5.3 Upgrading the Kernel

The kernel is already in the new JFFS2 image file when you upgrade the JFFS2.img file. However, if you wish to upgrade the kernel (pImage) without upgrading the JFFS2 image file, proceed as follows:

1. Ensure that both the development kit and the computer from which you wish to download the pImage file are connected to the Internet. The 10/100 Base T port is used to connect the development kit to the Internet.

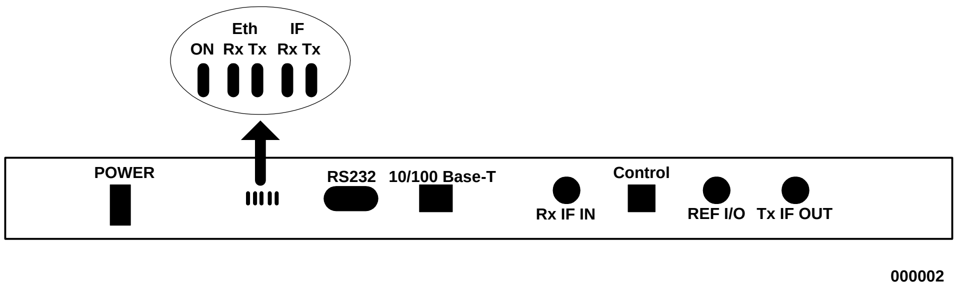


Figure 10 - Console Interface

2. Configure your computer as a FTP server.
3. Go into the directory named boot by typing the following command:

```
cd /boot ↵
```

4. Erase the pImage file that is currently in the flash memory by typing the following command:

```
rm pImage ↵
```

5. Get the new pImage file from Wavesat’s web site.
6. Save the new pImage file to the local FTP sever download directory.
7. Reset the development kit by restarting it or by typing the following command:

```
reboot ↵
```

-End-

5.4 Upgrading the FPGA File

If you need to upgrade the FPGA file, once the development kit is powered up, you can download it by performing an FTP operation as follows:

1. Ensure that both the development kit and the computer from which you wish to download the files from are connected to the Internet. The 10/100 Base T port is used to connect the development kit to the Internet.

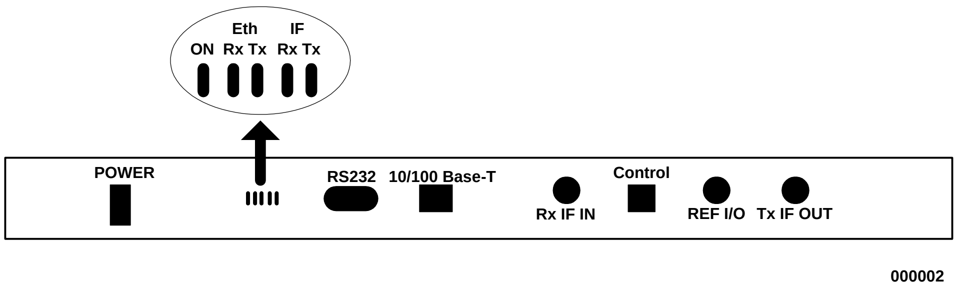


Figure 11 - Console Interface

2. Configure your computer as a FTP server.
3. Get the new FPGA file from Wavesat website.
4. Save the new FPGA load file to your FTP download directory.
5. Go to the directory named dragon by typing the following command:

```
cd /dragon ↵
```

6. Remove the FPGA file that is currently in the flash memory.

   For example, if the filename is top66zb6k_e.bin, remove the file as follows:

```
rm top66zb6k_e.bin ↵
```

-End-

7. Using vi editor, update the “base” script and the “sub” script so that they call the new FPGA file. (See Appendix A:vi Commands.)

   For example, if the filename is top66zb6k_e.bin, modify the following command as follows:

   ```
   ./fpga_prog top66zb6k_e.bin ↵
   ```

8. Reset the development kit by power-recycling the development kit or by typing the following command:

   ```
   reboot ↵
   ```

-End-

5.5 Upgrading the Command Interface Software

If you need to upgrade the command interface software, once the development kit is powered up, you can download it by performing an FTP operation as follows:

1. Ensure that both the development kit and the computer from which you wish to download the files from are connected to the Internet. The 10/100 Base T port is used to connect the development kit to the Internet.

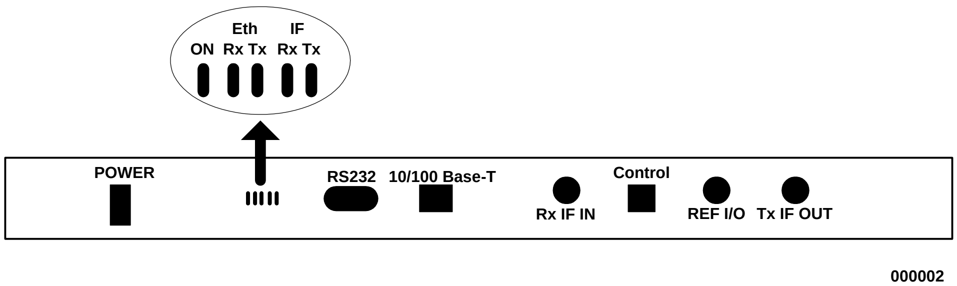


Figure 12 - Console Interface

2. Configure your computer as a FTP server.
3. Go into the directory named dragon by typing the following command:

```
cd /dragon ↵
```

4. Remove all files by typing the following command:

```
rm * ↵
```

Caution	This command, will remove all of the files in the directory including the FPGA loads.

5. Get all new software files, scripts and FPGA load from Wavesat's web site and save them to a local FTP server.
6. Connect to the local FTP server
7. To change the permissions on the files from read only to read write and execute, type:

```
chmod 777 * ↵
```

8. Reset the development kit by restarting it or by typing the following command:

```
reboot ↵
```

-End-

5.6 Programming the FPGA

To program the FPGA with the FPGA load that is currently residing on the Flash File System (JFFS), type:

```
./fpga_prog [fpga_load] ↵
```

Where

Parameter	Description
[fpga_load]	Name of the FPGA load

5.7 Programming the compressed FPGA

To program the DM 256 with a compressed file, type:

```
./fpga_prog [-c] fpga_load_file ↵
```

Note	The –c flag indicates the file is stored in a compressed file format .gz..

This page has been left intentionally blank.

Section: 6 End-to-End Test Connections

This section provides the necessary information for users who wish to perform or know the following:

- Number of Data Symbols to use.
- To add a Connection ID between the base station and a subscriber station.
- To remove a Connection ID between the base station and a subscriber station.
- To modify the modulation rates or the number or the number of symbols for an existing connection.
- To show the list of existing connections at base station.
- To show the connection IDs (CID) used by a subscriber station.

Note:	The Enter key is noted by the following symbol: ↵

6.1 Number of Data Symbols to Use

The maximum number of OFDM symbols that can be transmitted by a frame depends on the channel bandwidth and Cyclic Prefix (CP) size. The following table specifies the maximum number of data symbols that a frame can contain in Time Division Duplexing (TDD) and Frequency Division Duplexing (FDD) and at different frame duration codes. The chart is based on the following settings and assumptions:

- Transmit/Receive Transition Gap (TTG) time = 20 (applicable in TDD only).
- Receive/Transmit Transition Gap (RTG) time = 20 (applicable in TDD only).
- Burst #1Lenght = 14
- BPSK ½
- Nopreamble bits are set
- Contains fix-sized downlink and uplink maps.

Table 9 - Maximum Number of Data Symbols for Burst #2

Channel BW (MHz)	CP size	Frame Duration Code	Frame Duration (ms)	Downlink Allocation		Uplink Allocation	
				#Data Symbols (FDD)	#Data Symbols (TDD)	#Data Symbols(FDD)	#Data Symbols (TDD)
3.5	1/16	0	2.5	19	1	25	7
		1	4	41	12	47	18
		2	5	56	19	62	26
		3	8	100	41	106	48
		4	10	130	56	136	62
		5	12.5	166	74	172	81
		6	20	277	130	283	135

Note: Currently, due to a limitation of the CPU, the maximum is 255 symbols.

6.2 Add a Connection

To add a connection ID between the base station and a subscriber station, you need to perform the following:

- Add the connection ID (CID) at the base station.
- Add the connection ID (CID) at the subscriber station.

6.2.1 Add a Connection at the Base Station

To add a connection at the base station, type:

```
./macdb_ctrl add MAC_ADDRESS CID Type DL_Mod #DL_Sym UL_Mod #UL_Sym
Preamble Midamble ↵
```

Where

Parameter	Description
MAC_ADDRESS	Subscriber MAC Address
CID	Connection identifier (1 - 36)
Type	Transport or LinkTest
DL_Mod	Modulation rate to use in the downlink: Choices are: BPSK-1/2, , BPSK-3/4, QPSK-1/2, QPSK-3/4, QAM16-1/2, QAM16-3/4, QAM64-2/3, QAM 64-3/4
#DL_Sym	Number of data symbols to allocate in the downlink
UL_Mod	Modulation rate to use in the uplink: Choices are Choices are: BPSK-1/2, , BPSK-3/4, QPSK-1/2, QPSK-3/4, QAM16-1/2, QAM16-3/4, QAM64-2/3, QAM 64-3/4
#UL_Sym	Number of data symbols to allocate in the uplink
Preamble	Preamble (0 or 1).
Midamble	Midamble (0-3).

6.2.2 Add a Connection ID for received software-based PRBS at a Subscriber Station

To add a connection ID (CID) for received software based PRBS at a subscriber station or to filter out unwanted connections by the map processor, type:

```
./ss_linktest_ctrl add cid type ↵
```

Where

Parameter	Description
cid	Connection identifier (1 – 36)
type	Type of connection is either: Transport, LinkTest, or Broadcast

Note:	Currently Transport and Broadcast connection types are not available.

6.3 Delete a MAC Address

To remove a MAC Address at the base station you need to perform the following:

6.3.1 Delete a MAC Address at the Base Station

To delete a MAC Address at the base station, type:

```
./macdb_ctrl del MAC|CID arg ↵
```

Where

Parameter	Description
MAC|CID	Choose MAC for MAC Address.
arg	Enter the corresponding arguments MAC Address

6.4 Delete a Connection

To remove a connection ID between the base station and a subscriber station, you need to perform the following:

- Remove the connection ID (CID) at the base station.
- Remove the connection ID (CID) at the subscriber station.
- To remove a MAC Address at the base station you will need to perform only the

6.4.1 Delete a Connection ID at the Base Station

To delete a MAC Address or connection ID (CID) at the base station, type:

```
./macdb_ctrl del MAC|CID arg ↵
```

Where

Parameter	Description
MAC|CID	Choose either CID for Connection Identifier to be deleted.
arg	Enter the corresponding arguments: Connection identifier (1 - 36)

6.4.2 Delete a Connection ID at a Subscriber Station

To delete a connection ID (CID) at a subscriber station, type:

```
./ss_linktest_ctrl del cid↵
```

Where

Parameter	Description
cid	Connection identifier (1 - 36)

6.5 Modify a Connection

The following parameters can be modified for an existing connection:

- Modulation rates
- Number of symbols

This operation can be performed at the base station only. To modify an existing connection, type:

```
./macdb_ctrl modify CID DL_Modulation #DL_Symbols UL_Modulation
#UL_Symbols Preamble Midamble ↵
```

Note:	Ensure that you have typed in the entire command until the symbol ↵ where you would press the enter key.

Where

Parameter	Description
CID	Connection identifier (1 - 36)
DL_Modulation	New modulation rate to use in the downlink: Choices are: BPSK-1/2, , BPSK-3/4, QPSK-1/2, QPSK-3/4, QAM16-1/2, QAM16-3/4, QAM64-2/3, QAM 64-3/4
#DL_Symbols	New number of data symbols to allocate in the downlink
UL_Modulation	New modulation rate to use in the uplink: Choices are: BPSK-1/2, , BPSK-3/4, QPSK-1/2, QPSK-3/4, QAM16-1/2, QAM16-3/4, QAM64-2/3, QAM 64-3/4
#UL_Symbols	New number of data symbols to allocate in the uplink
Preamble	Preamble (0 or 1).
Midamble	Midamble (0-3).

6.6 Show the List of Connections

To show the list of existing connections, type the following at the base station development kit only:

```
./macdb_ctrl show ↲
```

The console returns the following:

```
MAC Address    SS State   CID State    DL Mod      DL Sym  UL Mod      UL
Sym     Pre Mid

LinkTest Connections:

CID DL Mod   DL Sym UL Mod   UL Sym Pre Mid
```

Where

Parameter	Description
MAC Address	Subscriber MAC Address
SS State	Subscriber Station State (currently not used)
CID	Connection identifier (1 - 36)
DL_Mod	Modulation rate used for the data symbols in the downlink
#DL_Sym	Number of data symbols allocated in the downlink per frame.
UL_Mod	Modulation rate used for the data symbols in the uplink
#UL_Sym	Number of data symbols allocated in the uplink per frame
Preamble	Preamble (0 or 1).
Midamble	Midamble (0-3).

6.7 Showing the Connection IDs used by a Subscriber Station

To show the connection IDs (CID) and the connection type used by a subscriber station, type the following at the subscriber station:

```
./ss_linktest_ctrl show ↵
```

The console returns the following:

```
CID Type
```

Where

Parameter	Description
CID	List of connections
Type	Type of connection is either Transport, LinkTest or Broadcast.

This page has been left intentionally blank.

Section: 7 Buffer Read/Write Operations

This section provides the necessary information for users who wish to perform the following:

- To read the content of Tx Frame Configuration Buffer 1 or 2.
- To read the content of Rx Frame Configuration Buffer 1 or 2.
- To write to the Tx Payload Buffer.
- To read from the Rx Payload Buffer.
- To reset the Tx Payload Buffer.
- To reset the Rx Payload Buffer.

Note:	The Enter key is noted by the following symbol: ↵

7.1 Read a Frame Configuration Buffer

7.1.1 Read the content of Tx Frame Configuration Buffer 1

To read the content of Tx Frame Configuration Buffer 1, type:

```
./phy_ctrl get tx_cfg_buff1 value↵
```

Where

Parameter	Description
value	The value is the length in numbers of words (32 bits) to be read.

The console displays 152 bits of configuration data at a time as follows:

```
150058C1-09004EA0-4801000A-2E000701-00099000-0AA00004-
00000000-00000000-00000000-00000000-00000000-00000000-
00000000-00000000-00000000-00000000-00000000-00000000-
00000000-00000000-00000000-00000000-00000000-00000000-
00000000-00000000-00000000-00000000-00000000-00000000-
00000000-00000000-00000000-00000000-00000000-00000000-
00000000-00000000-
```

7.1.2 Read the content of Tx Frame Configuration Buffer 2

To read the content of Tx Frame Configuration Buffer 2, type:

```
./phy_ctrl get tx_cfg_buff2 value↵
```

Where

Parameter	Description
value	The value is the length in numbers of words (32 bits) to be read.

The console displays 152 bits of configuration data at a time as follows:

```
150058C1-09004EA0-4801000A-2E000701-00099000-0AA00004-
00000000-00000000-00000000-00000000-00000000-00000000-
00000000-00000000-00000000-00000000-00000000-00000000-
00000000-00000000-00000000-00000000-00000000-00000000-
00000000-00000000-00000000-00000000-00000000-00000000-
00000000-00000000-00000000-00000000-00000000-00000000-
00000000-00000000-
```

7.1.3 Read the content of Rx Frame Configuration Buffer 1

To read the content of Rx Frame Configuration Buffer 1, type:

```
./phy_ctrl get rx_cfg_buff1 value↵
```

Where

Parameter	Description
value	The value is the length in numbers of words (32 bits) to be read.

The console displays 152 bits of configuration data at a time as follows:

```
150058C1-09004EA0-4801000A-2E000701-00099000-0AA00004-
00000000-00000000-00000000-00000000-00000000-00000000-
00000000-00000000-00000000-00000000-00000000-00000000-
00000000-00000000-00000000-00000000-00000000-00000000-
00000000-00000000-00000000-00000000-00000000-00000000-
00000000-00000000-00000000-00000000-00000000-00000000-
00000000-00000000-
```

7.1.4 Read the content of Rx Frame Configuration Buffer 2

To read the content of Rx Frame Configuration Buffer 2, type:

```
./phy_ctrl get rx_cfg_buff2 value↵
```

Where

Parameter	Description
value	The value is the length in numbers of words (32 bits) to be read.

The console displays 152 bits of configuration data at a time as follows:

```
150058C1-09004EA0-4801000A-2E000701-00099000-0AA00004-
00000000-00000000-00000000-00000000-00000000-00000000-
00000000-00000000-00000000-00000000-00000000-00000000-
00000000-00000000-00000000-00000000-00000000-00000000-
00000000-00000000-00000000-00000000-00000000-00000000-
00000000-00000000-00000000-00000000-00000000-00000000-
00000000-00000000-
```

7.2 Write to the Tx Payload Buffer

To write to the DM256 Tx payload buffer, type:

```
./phy_ctrl set tx_fifo value ↵
```

Where

Parameter	Description
value	1 byte (8 bits) of data.

7.3 Read to the DM 256 Rx Payload Buffer

To read the next 8 bits from the DM256 Rx payload buffer, type:

```
./phy_ctrl get rx_fifo ↵
```

The console returns a byte value. However, if the DMA transfer width (default mode is byte mode) is less than one word, then only the most significant bits (MSB) of the value are used. If the DMA transfer width is 1 byte then only the 8 MSB are used

Note:	The DMA transfer width (number of bits to transfer) is set by the CPU.

7.4 Reset the Tx Payload Buffer

To clear the content of the Tx payload buffer, type:

```
./phy_ctrl set tx_data_fifo_sync_reset value ↵
```

Where

Parameter	Description
value	0: Normal setting 1: The buffer is cleared

To show the content of the Tx payload buffer, type:

```
./phy_ctrl get tx_data_fifo_sync_reset ↵
```

7.5 Reset the Rx Payload Buffer

To clear the content of the Rx payload buffer, type:

```
./phy_ctrl set rx_data_fifo_sync_reset value ↵
```

Where

Parameter	Description
value	0: Normal setting 1: The buffer is cleared

To read the content of the Rx payload buffer, type:

```
./phy_ctrl get rx_data_fifo_sync_reset ↵
```

7.6 Enable and Disable Serial Port and Serial Mode

The following table indicates the function when serial port and/or serial mode are enabled or disabled:

Serial mode	Serial port	Configuration
0	0	The serial port is disabled (normal setting). The Tx payload buffer transmits parallel data to the CPU and the Rx payload buffer receives parallel data from the CPU.
0	1	Behavior undefined (do not use this configuration)
1	0	The serial port is emulated. This configuration serves to test the serial port module. In this mode, the payload from the MAC Layer is serialized before it is fed to the Tx serial port modules. In the Rx channel, the data received is also serialized and fed to the Rx serial port modules as if it is coming from the serial port.
1	1	The serial port is enabled. The Tx payload buffer transmits data to the serial port and the Rx payload buffer receives data from the serial port.

The below commands can be applied to both receive and transmit at the same time.

7.6.1 Serial Mode

To enable the serial mode, which allows the use of logic for the serial mode, type:

```
./phy_ctrl set serial_mode value ↵
```

Where

Parameter	Description
`value`	0: Normal setting 1: Enable serial mode

To display the status of the serial mode, type:

```
./phy_ctrl get serial_mode ↵
```

7.6.2 Serial Port

To enable the serial port, type:

```
./phy_ctrl set serial_port value ↵
```

Where

Parameter	Description
`value`	0: Normal setting 1: Enable serial port

To display the status of the serial mode, type:

```
./phy_ctrl get serial_port ↵
```

This page has been left intentionally blank.

Section: 8 Link Monitoring

This section provides the necessary information for users who wish to perform the following:

- To verify whether the Rx link is currently active.
- To verify whether the Rx link is currently configured.
- To display the state of the Tx payload buffer – Tx Data Ready
- To display the state of the Rx payload buffer – Rx Data Ready.
-

> Note: The Enter key is noted by the following symbol: ↵

8.1 Verify Rx Link Active

To verify whether the Rx link is currently active or not, type:

```
./phy_ctrl get rx_link_active ↵
```

The console displays the setting in decimal and hexadecimal values:

```
rx_link_active -> Value: 0, Hex value: 0x00000000
```

Where

Parameter	Description
`Value`	0: Rx link is currently not active (No preamble is detected.) 1: Rx link is currently active (At least one preamble detected in the last two frames.)

8.2 Verify Rx Link Configured

To verify whether the Rx link is currently configured or not, type:

```
./phy_ctrl get rx_link_config ↵
```

The console displays the setting in decimal and hexadecimal values:

```
rx_link_config -> Value: 0, Hex value: 0x00000000
```

Where

Parameter	Description
`Value`	0: Rx link is currently not configured (FCH not detected in incoming frame.) 1: Rx link is currently configured (FCH detected in incoming frame.)

8.3 Display Tx Data Ready- Tx Payload Buffer

To display the state of the Tx payload buffer interrupts, type:

```
./phy_ctrl get tx_data_ready ↵
```

The console displays the setting in decimal and hexadecimal values:

```
tx_data_ready -> Value: 1, Hex value: 0x00000001
```

Where

Parameter	Description
Value	0: The buffer is more than 1/4 full and is not ready to accept data. 1: The buffer is 1/4 full or less, and is ready to accept data.

8.4 Display Rx Data Ready – Rx Payload Buffer

To display the state of the Rx payload buffer interrupt, type:

```
./phy_ctrl get rx_data_ready ↵
```

The console displays the setting in decimal and hexadecimal values:

```
rx_data_ready -> Value: 0, Hex value: 0x00000000
```

Where

Parameter	Description
Value	0: The buffer is less than 3/4 full and is not ready to release data. 1: The buffer is 3/4 full or more, and is ready to release data.

Section: 9 Performance Statistics

The following subsections describe the commands used to query the status of the link test connection (CID = 10) or the broadcast connection (CID = 37).

- Number of Data Symbols to use.
- To add a Connection ID between the base station and a subscriber station.
- To remove a Connection ID between the base station and a subscriber station.
- To modify the modulation rates or the number of symbols for an existing connection.
- To show the list of existing connections at base station.
- To show the connection IDs (CID) used by a subscriber station.
- To show the Signal to Noise Ratio and Power Level on symbol following REF2 symbol.

Note:	The Enter key is noted by the following symbol: ↵

9.1 Obtaining the Packet Statistics for a Connection

To obtain packet statistics for a given connection, type:

```
./ofdm_stats data cid ↵
```

Where

Parameter	Description
cid	Connection identifier (1 - 36)

The console displays the following statistics:

```
Good: 0, Bad: 0, Bytes: 00, HeadErr: 0, Buff overflow: 0
```

Where

Parameter	Description
Good	Cumulative number of error-free packets received (PDU)
Bad	Cumulative number of packets received that contain errors meaning that the packet data failed the parity check (CRC16) (PDU)
Bytes	Cumulative number of bytes received
HeadErr	Cumulative number of MAC headers that failed the parity check (CRC8)
Buff overflow	Cumulative number of buffer overflows (FIFO Full)

9.2 Obtaining the BER Statistics for a Connection

To obtain the BER measurements for a connection, type:

```
./ofdm_stats ber cid ↵
```

Where

Parameter	Description
cid	Connection identifier (1 - 36)

The console displays the following statistics:

```
Cid: 00, Nb bits: 00, Nb errors: 00, Nb occur: 00, Sync errors: 0, Lost
packet: 0
```

Where

Parameter	Description
Cid	Connection identifier (1 - 36)
Nb bits	Cumulative number of good bits
Nb errors	Cumulative number of bits in error
Nb occur	Cumulative number of occurrences of bits in error
Sync errors	Cumulative number of synchronization errors. A synchronization error occurs when the number of errors within a packet is too many such that synchronization must be re-attempted.
Lost packet	Cumulative number of packets where the sequence number was not consecutive

9.3 Obtaining the Link Quality Statistics for a Connection

To obtain link quality statistics for a connection, type:

```
./ofdm_stats link cid↵
```

Where

Parameter	Description
cid	Connection identifier (1 - 36)

The console displays the following statistics:

```
Cid: 10, snr :0.00, RangeDelay: 0, freq_corr_angle: 0, corr_rs_bit: 0,
corr_viterbi_bit: 0, avrgPower: 0.00, integer_offset: 0
```

Where

Parameter	Description
Cid	Connection identifier (1 - 36)
snr	Average signal power to noise ratio. The snr value returned is between 0-63.75 dB (in increments of 0.25 dB) and must be divided by 4 in order to obtain the actual snr value. actual snr = returned snr value / 4
RangeDelay	Applicable only at the base station. It is a measure of the point in time when the first sample of the preamble is detected relative to the sample time during which the burst was scheduled. This is used for initial ranging and periodic ranging processes. If the Range Delay is not 0 ($^{+}$1 or -1) then set the ./phy_ctrl set delay_connection at the substation.)
Freq_corr_angle	16-bit signed number providing the frequency offset, up to +/-1/2 the subcarrier spacing; the frequency offset between the subscriber station and the base station is given by (fs/256)*(freq_corr_angl/256) where fs is the baseband sampling clock (e.g. fs =40 MHz for the 3.5 MHz channel bandwidth).
Corr_rs_bit	Cumulative number of bytes corrected by RS FEC
Corr_viterbi_bit	Cumulative number of bits corrected by Viterbi FEC
AvrgPower	Power measurement (in millivolts) of the Reference symbol
Integer_offset	This parameter represents the offset of the subcarrier frequency.

9.4 Resetting the Statistical Measurements

To reset the statistical data, type:

```
./ofdm_stats reset all | cid cid_value ↵
```

Where

Parameter	Description
all|cid	Use either of the following values: "all" to reset the statistics for all connections to zero.
cid_value	Or reset Connection identifier (1 - 36) to zero or cid_value (0-37)

9.5 Obtaining Specific Symbol Statistics

To obtain specific link quality statistics on a particular symbol, type:

```
./ofdm_stats symbol REF1_CELL | CONFIG_CELL | IDLE_CELL | REF2_CELL |
FCH_CELL |MIDAMBLE_CELL | GAP_CELL | DATA_CELL ↵
```

Where

Parameter	Description
REF1_CELL | CONFIG_CELL | IDLE_CELL | REF2_CELL | FCH_CELL MIDAMBLE_CELL | GAP_CELL | DATA_CELL	Use one of the following values: REF1_CELL, CONFIG_CELL , IDLE_CELL, REF2_CELL, FCH_CELL, MIDAMBLE_CELL, GAP_CELL. or DATA_CELL to obtain specific statistical information. Certain symbol types are received only when the DM 256 is placed in certain modes: i.e. pdu_passthrough =1. REF1_CELL pdu_passthrough =1 REF2_CELL pdu_passthrough =1 FCH_CELL pdu_passthrough =1 MIDAMBLE_CELL pdu_passthrough =1 GAP_CELL pdu_passthrough =1 DATA_CELL pdu_passthrough =1 or 0 CONFIG_CELL, IDLE_CELL pdu_passthrough N/A

9.6 Signal to Noise Ratio and Power Level Statistics

To show the Signal to Noise Ratio and Power Level on symbol following REF2 symbol, type:

```
./phy_ctrl get snr_power ↵
```

Note that this command returns 4X the actual value.

$$\mathrm{SNR\ (db)} = \frac{(\mathrm{snr - power})}{4}$$

Section: 10 Calibration and Testing

The following sections describe the how to conduct calibration and testing of the development kit:

- How to generate PRBS-23 Data for BER (Bit Error Rate) Testing.
- How to set the Rx Noise Generator.
- How to obtain the Tx Gain Calibration Curve
- How to obtain the Tx DAC Calibration Curve
- How to determine the optimal Tx digital gain.
- How to determine the optimal Rx gain.
- How to obtain the BER vs SNR (Signal to Noise Ratio) curve for a given profile.
- How to verify development kit's performance in different Stanford University Interim (SUI) channel models.
- How to conduct loopback testing using raw data loopback, packet loopback, encoded word loopback, and sample feedback.

Note:	The Enter key is noted by the following symbol: ↵

10.1 Generating PRBS-23 Data

This function is to simulate the BER generator function (PRBS 23) of a BER tester.

To set the PRBS 23 data generator, type:

```
./phy_ctrl set tx_rand_stream value↵
```

Where

Parameter	Description
`value`	0: The PRBS 23 data generator function is disabled (Normal setting) 1: The PRBS 23 data generator function is enabled

To show the PRBS 23 data generator, type:

```
./phy_ctrl get tx_rand_stream ↵
```

10.2 Setting the Rx Noise Generator

This function is to set the Rx Noise generator function.

To set the Rx Noise Generator, type:

```
./phy_ctrl set rx_noise_inducer value↵
```

Where

Parameter	Description
`value`	0: The Rx Noise Generator is disabled (Normal setting) 1: The Rx Noise Generator is enabled.

To show the whether the Rx Noise generator has been enabled or not, type:

```
./phy_ctrl get rx_noise_inducer↵
```

10.3 Obtaining the Tx Gain Calibration Curve

Use this procedure to obtain the calibration curve of the Tx gain. The calibration curve is obtained by varying the Tx gain setting and measuring the signal power transmitted by the development kit for each of these settings.

Note:	This operation must be performed before any other calibration or test.

This procedure can be performed using the following settings:

- Profile: QPSK-1/2 (this procedure can be repeated for the other profiles).
- CID: 10 (any CID between 1 and 36 can be used).
- Downlink symbols: 63; uplink symbols: 0 (to fill up the downlink sub-frame).
- This is dependant upon: Bandwidth = 3.5, CP Size = 1/16, Frame Duration = 2, Transmission method = FDD.

Set the equipment as follows:

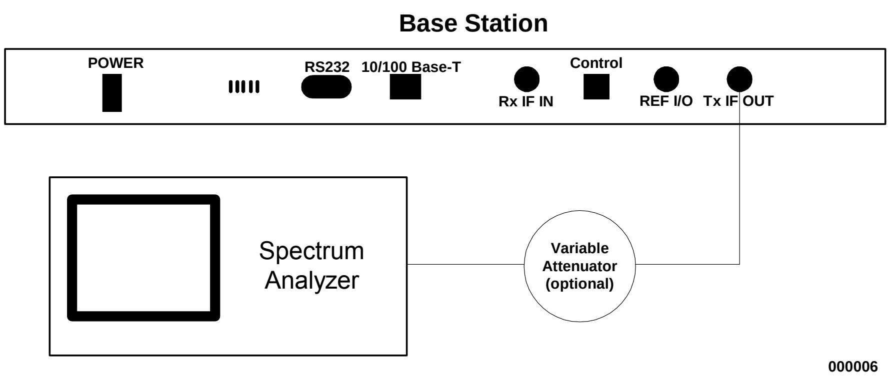


Figure 13 - Equipment Setup for Tx Gain Calibration

1. Configure the development kit by executing the following script, type:

   ```
   ./base ↵
   ```

2. Add a connection by typing the following command:

   ```
   ./macdb_ctrl add 00:00:00:00:00:00 10 LinkTest QPSK-1/2 63 QPSK-1/2 0 0 0↵
   ```

3. Adjust the Spectrum Analyzer so that it can display the full range of the signal power. If necessary, reduce the signal power using the attenuator.

   Center Frequency = 10 MHz,

   Span = 10 MHz,

   Reference = 0 dBm

-Continue-

4. Set the Tx gain to a value between 0 and 255 by typing the following command:

```
./phy_ctrl set tx_gain_value value↵
```

5. Record the signal power obtained with the Spectrum Analyzer.
6. Repeat steps 4 and 5 at different Tx gain settings.
7. Plot a graph of the Tx signal power measured from the spectrum analyzer versus the Tx gain settings.

-End-

10.4 Obtaining the Tx DAC Calibration Curve

Use this procedure to obtain the calibration curve of the Tx Digital-to-Analog Converter (DAC). The calibration curve is obtained by varying the Tx DAC setting and measuring the signal power transmitted by the development kit for each of these settings.

Note:	This operation must be performed after the Tx Gain Calibration Curve (previous procedure) but before any other calibration or test. Make sure that the Tx gain is set to default value.

This procedure can be performed using the following settings:

- Profile: QPSK-1/2 (this procedure can be repeated for the other profiles).
- CID: 10 (any CID between 1 and 36 can be used).
- Downlink symbols: 63; uplink symbols: 0 (to fill up the downlink sub-frame).
- This is dependant upon: Bandwidth = 3.5, CP Size = 1/16, Frame Duration = 2, Transmission method = FDD.

Set the equipment as follows:

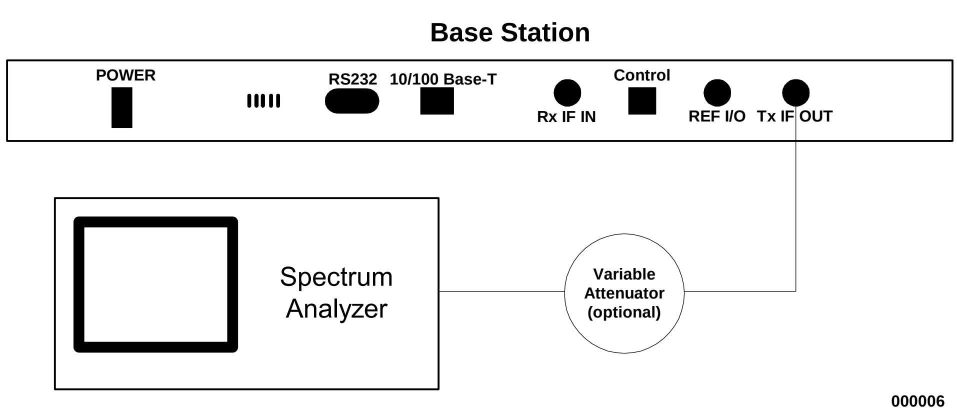


Figure 14 - Equipment Setup for Tx DAC Calibration

1. Configure the development kit by executing the following script, type:

   ```
   ./base ↵
   ```

2. Add a connection by typing the following command:

   ```
   ./macdb_ctrl add 00:00:00:00:00:00 10 LinkTest QPSK-1/2 63 QPSK-1/2 0 0 0↵
   ```

3. Adjust the Spectrum Analyzer so that it can display the full range of the signal power. If necessary, reduce the signal power using the attenuator.
4. Set the DAC to a value between 0 and 255 by typing the following command:

   ```
   ./phy_ctrl set spi_tx_dac value↵
   ```

5. Record the signal power obtained with the Spectrum Analyzer.
6. Repeat steps 4 and 5 at different DAC settings.
7. Plot a graph of the Tx signal power versus the DAC settings to obtain the calibration curve of the Tx DAC.

   An example of the graph is shown in "Appendix B: Calibration and Test Results".

-End-

10.5 Determining the Optimal Tx Digital Gain

Use this procedure to obtain the optimal Tx digital gain. This is obtained by determining the Tx digital gain value that yields the highest Signal-to-Noise Ratio (SNR) while keeping the Rx power constant. This procedure can be performed using the following settings:

- Profile: QPSK-1/2 (this procedure can be repeated for the other profiles).
- CID: 10 (any CID between 1 and 36 can be used).
- Downlink symbols: 63; uplink symbols: 0 (to fill up the downlink sub-frame).
- This is dependant upon: Bandwidth = 3.5, CP Size = 1/16, Frame Duration = 2, Transmission method = FDD.
- Recommended Tx DAC: 90 (or the optimal value obtained in the Tx DAC calibration).
- Recommended Rx DAC: 105.
- Rx power (obtained from the "./ofdm_stats link command"): approximately 45 dB.
- Tx scaling: -1 (Setting Tx scaling to -1 prevents saturation. Setting Tx scaling to a value different than -1 will cause a degradation of the SNR, -1 is the optimal value.).

Note:	This operation must be performed after the Tx Gain Calibration Curve and the Tx DAC Calibration Curve calibrations.

Set up the equipment as shown:

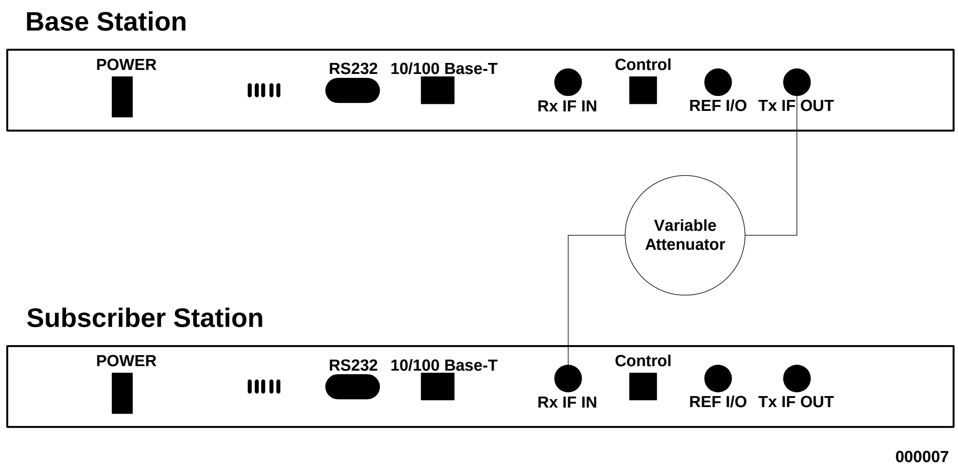


Figure 15 - Equipment Setup Determination of Optimal Tx Digital Gain

1. Set the base station development kit as follows:
   - Configure the development kit by executing the following script, type:

     ```
     ./base ↵
     ```

   - Add a connection by typing the following command:

     ```
     ./macdb_ctrl add 00:00:00:00:00:00 10 LinkTest QPSK-1/2 63 QPSK-1/2 0 0 0↵
     ```

   - Set the Tx DAC to the 90 by typing the following command:

     ```
     ./phy_ctrl set spi_tx_dac 90 ↵
     ```

2. Set the subscriber station development kit as follows:
   - Configure the development kit by executing the following script, type:

     ```
     ./sub ↵
     ```

   - Set the Rx DAC to 105 by typing the following command:

     ```
     ./phy_ctrl set spi_rx_dac 105 ↵
     ```

3. Set the Tx gain to a value between 0 and 255 by typing the following command:

   ```
   ./phy_ctrl set tx_gain_value value↵
   ```

4. Display the Rx power (AvrgPower) in dB at the subscriber station by typing the following command:

   ```
   ./ofdm_stats link 10 ↵
   ```

5. Adjust the attenuator to set the Average Receive Power to approximately 55 dB.
6. Display the Rx power (AvrgPower) and the SNR by typing the following command:

   ```
   ./ofdm_stats link 10 ↵
   ```

7. Record the two measurements.
8. Repeat steps 3 to 7 at different Tx Gain settings.
9. Plot a graph of the SNR versus the Tx Gain settings. The optimal Tx digital gain is where the SNR is highest. An example of the graph is shown in “Appendix B:Calibration and Test Results”.

-End-

10.6 Determining the Optimal Rx Power

Use this procedure to obtain the optimal Rx power. This procedure can be performed using the following settings:

- Profile: QPSK-1/2 (this procedure can be repeated for the other profiles).
- CID: 10 (any CID between 1 and 36 can be used).
- Downlink symbols: 63; uplink symbols: 0 (to fill up the downlink sub-frame).
- This is dependant upon: Bandwidth = 3.5, CP Size = 1/16, Frame Duration = 2, Transmission method = FDD.
- Recommended Tx DAC: 90 (or the optimal value obtained in the Tx DAC calibration).
- Recommended Rx DAC: 105.

Note:	This operation must be performed after the Tx Gain Calibration Curve and the Tx DAC Calibration Curve calibrations.

Set up the equipment as follows:

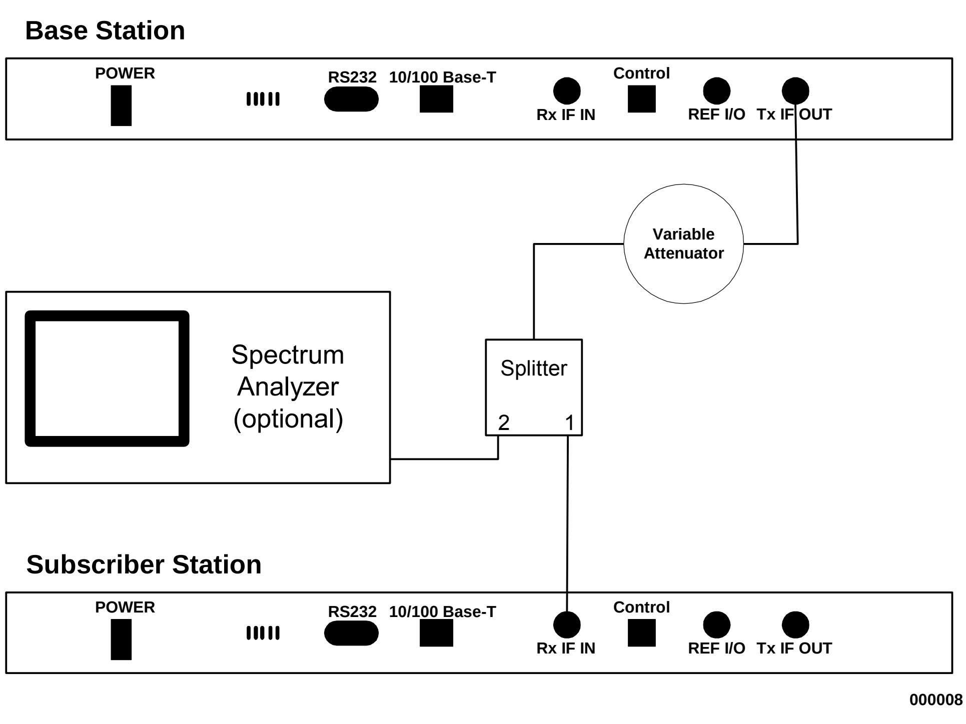


Figure 16 - Equipment Setup Determination of Optimal Rx Power

1. Set the base station development kit as follows:

   - Configure the development kit by executing the following script, type:

     `./base ↵`

   - Add a connection by typing the following command:

   `./macdb_ctrl add 00:00:00:00:00:00 10 LinkTest QPSK-1/2 63 QPSK-1/2 0 0 0↵`

   - Set the Tx DAC to 90 by typing the following command:

     `./phy_ctrl set spi_tx_dac 90 ↵`

2. Set the subscriber station development kit as follows:

   - Configure the development kit by executing the following script, type:

     `./sub ↵`

   - Set the Rx DAC to 105 by typing the following command:

     `./phy_ctrl set spi_rx_dac 105 ↵`

3. Display the power and SNR of the receive signals by typing the following command:

   `./ofdm_stats link 10 ↵`

4. Vary the attenuator and record the Rx power (AvrgPower) in dB and SNR at different settings.
5. Plot a graph of the SNR versus the Rx power. The optimal Rx power is where the SNR is highest.

   An example of the graph is shown in "Appendix B:Calibration and Test Results".

-End-

10.7 BER Testing

Use this command to place the scheduler in a mode of operation to facilitate testing with a PRBS Analyzer used for coded BER Testing, type:

```
./sched_ctrl set bert value ↵
```

Where

Parameter	Description
`value`	0: Normal setting 1: Set in coded BER test mode

To show the mode of BERT testing, type:

```
./sched_ctrl get bert ↵
```

10.7.1 Uncoded BER Testing

Use this procedure to obtain the BER versus SNR curve for a given profile. This procedure can be performed using the following settings:

- Profile: BPSK-1/2, BPSK-3/4 , QPSK-1/2, QPSK-3/4, QAM16-1/2, QAM16-3/4, QAM64-2/3, QAM64-3/4 (this procedure can be repeated for any of these profiles).
- CID: 37 (Broadcast).
- Downlink symbols: Value to be determined using Table 10 - ; uplink symbols: 0 (to fill up the downlink sub-frame).
- This is dependant upon: Bandwidth, CP Size, and Frame Duration and Transmission Method = FDD.

Note:	This operation must be performed after the Tx Gain Calibration Curve and the Tx DAC Calibration Curve calibrations.

Set up the equipment as follows:

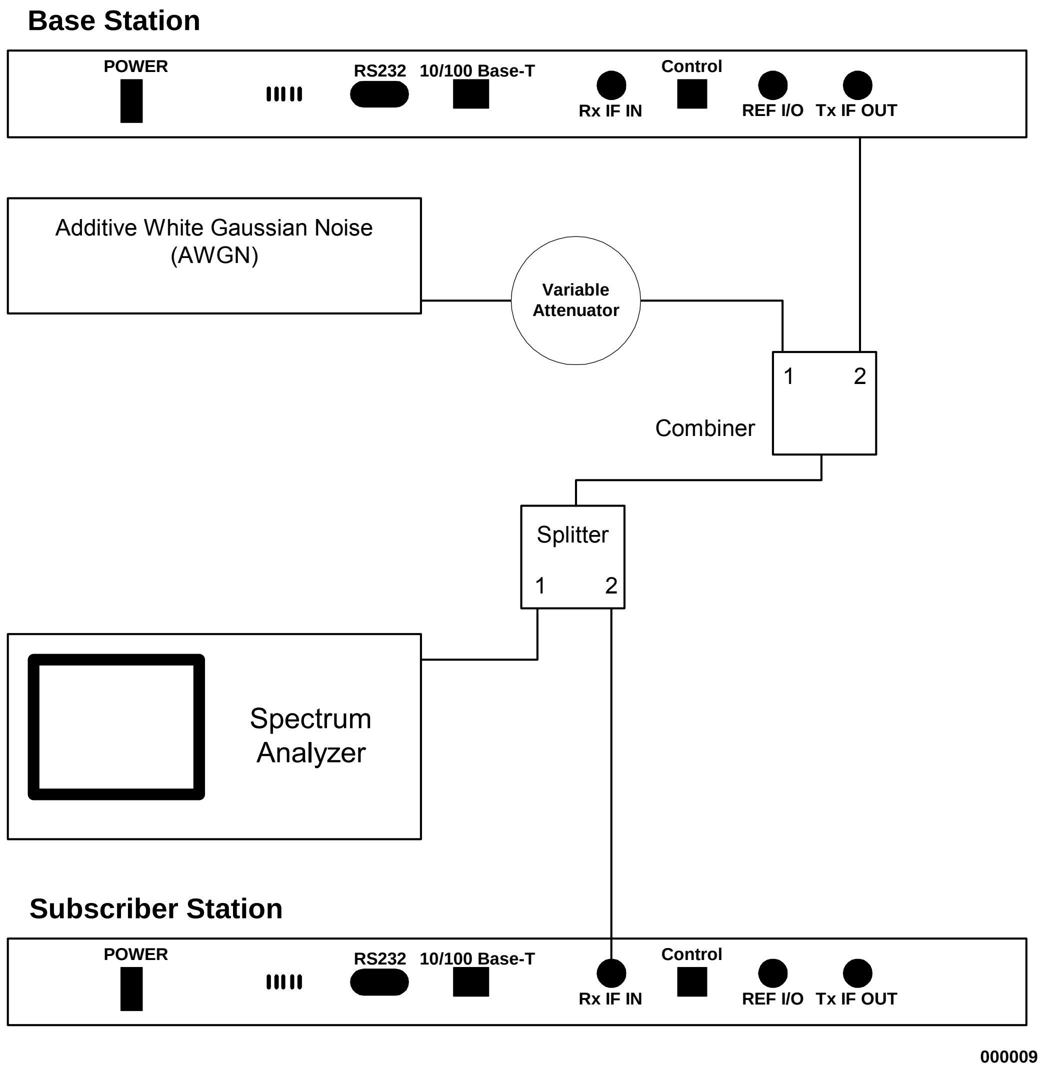


Figure 17 - Equipment Setup for Uncoded BER Test

1. Set the base station development kit as follows:

- Configure the development kit by executing the following script, type:

```
./base ↵
```

- Setting up a connection by typing the following command:

```
./sched_ctrl user modify 37 [DL_Modulation] [DL#_Allocation] QPSK-1/2 0 0 0 ↵
```

Where

Parameter	Description
DL_Modulation	BPSK-1/2, BPSK-3/4, QPSK-1/2, QPSK-3/4, QAM16-1/2, QAM16-3/4, QAM64-2/3, QAM64-3/4
DL#_Allocation	The number of symbols in a frame for a specific DL Modulation excluding the 3 overhead symbols. (See Table below helps determine the maximum number of symbols in a frame for each specific Frame Duration Code).

Table 10 - # of Symbols/Frame with Different Frame Duration Codes in FDD Mode

Channel BW (MHz)	CP size	Frame Duration 2.5 (ms) for Frame Duration Code 0 #Symbols/ Frame	Frame Duration 4.00 (ms) for Frame Duration Code 1 #Symbols/ Frame	Frame Duration 5.00 (ms) for Frame Duration Code 2 #Symbols/ Frame	Frame Duration 8.00 (ms) for Frame Duration Code 3 #Symbols/ Frame	Frame Duration 10.00 (ms) for Frame Duration Code 4 #Symbols/ Frame	Frame Duration 12.50 (ms) for Frame Duration Code 5 #Symbols/ Frame	Frame Duration 20.00 (ms) for Frame Duration Code 6 #Symbols/ Frame
	1/4	12.00	22.00	28.00	47.00	59.00	75.00	122.00
	1/8	14.00	24.00	31.00	52.00	66.00	83.00	135.00
	1/16	15.00	26.00	33.00	55.00	70.00	88.00	144.00
1.75	1/32	15.00	27.00	34.00	57.00	72.00	91.00	148.00
	1/4	28.00	47.00	59.00	97.00	122.00	153.00	247.00
	1/8	31.00	52.00	66.00	108.00	135.00	170.00	274.00
	1/16	33.00	55.00	70.00	114.00	144.00	180.00	291.00
3.5	1/32	34.00	57.00	72.00	118.00	148.00	186.00	300.00
	1/4	59.00	97.00	122.00	197.00	247.00	309.00	497.00
	1/8	66.00	108.00	135.00	219.00	274.00	344.00	552.00
	1/16	70.00	114.00	144.00	232.00	291.00	364.00	585.00
7	1/32	72.00	118.00	148.00	239.00	300.00	375.00	603.00
	1/4	86.00	139.00	175.00	282.00	354.00	443.00	711.00
	1/8	96.00	155.00	195.00	314.00	393.00	492.00	790.00
	1/16	102.00	165.00	207.00	333.00	417.00	522.00	837.00
10	1/32	105.00	170.00	213.00	343.00	429.00	537.00	862.00

000039

Note for Channel BW 10 MHz:	When the D/A sampling clock for the DM256 is 40 MHz, the baseband sampling rate for the 10 Mhz channelization is Fs = 11.424 MHz, which is not the rate specified in the P802.16-2004 standard. To meet that requirement the D/A sampling (and A/D sampling clock) needs to be 40.32 MHz

- Pause the transmission by typing the following command:

```
./sched_ctrl set pause 1 ↵
```

-Continue-

2. Set the subscriber station development kit as follows:

   - Configure the development kit by executing the following script, type:

     ```
     ./sub ↵
     ```

   - Reset the statistics readings by typing the following command:

     ```
     ./ofdm_stats reset all ↵
     ```

3. Resume the transmission at the base station by typing the following command:

   ```
   ./sched_ctrl set pause 0 ↵
   ```

4. Set the noise level using the attenuator to where the SNR is high enough so the development kit is barely decoding any data, to allow the signal to be read and low enough to generate corrected errors.
5. Pause the transmission at the base station by typing the following command:

   ```
   ./sched_ctrl set pause 1 ↵
   ```

6. Reset the statistics readings by typing the following command:

   ```
   ./ofdm_stats reset all ↵
   ```

7. Resume the transmission at the base station by typing the following command:

   ```
   ./sched_ctrl set pause 0 ↵
   ```

Note:	Wait for the Base Station and the Substation to transmit and receive Data.

8. Pause the transmission at the base station by typing the following command:

   ```
   ./sched_ctrl set pause 1 ↵
   ```

9. Record the accumulated number of good bits (Nb bits) after typing the following command:

   ```
   ./ofdm_stats ber 37 ↵
   ```

10. Record the SNR, the accumulated number of bytes corrected by RS FEC (Corr_rs_bit) and the accumulated number of bytes corrected by Viterbi FEC (Corr_viterbi_bit) after typing the following command:

    ```
    ./ofdm_stats link 37 ↵
    ```

11. Repeat steps 6 to 10 at different SNR's.

-Continue-

12. Calculate the BER at each noise level using the following equation:

BER: $$\frac{\text{Corr\_rs\_bit} + \text{Corr\_viterbi\_bit}}{\text{Nb bits}}$$

Note:	This will provide a very rough estimation of the BER.

13. Plot a graph of the BER versus the SNR.

An example of the graph is shown in "Appendix B:Calibration and Test Results".

The standard deviation of the BER values can be calculated using the following equation (from the derivative method of error estimation):

$$\text{Error} = \frac{\left(\text{Nb bits}^2 + 1\right)}{\text{Nb bits}^3}\left(\text{Corr\_rs\_bit} + \text{Corr\_viterbi\_bit}\right)$$

Note:	This should be done using a BER tester for a real BER curve.

-End-

10.7.2 Coded BER Testing

Use this procedure to obtain the BER versus SNR curve for a given profile. This procedure can be performed using the following settings:

- Profile: BPSK-1/2, BPSK-3/4, QPSK-1/2, QPSK-3/4, QAM16-1/2, QAM16-3/4, QAM64-2/3, QAM64-3/4 (this procedure can be repeated for any of these profiles).
- CID: 37 (Broadcast).
- Downlink symbols: Value to be determined using Table 11 - ; uplink symbols: 0 (to fill up the downlink sub-frame).
- This is dependant upon: Bandwidth, CP Size, and Frame Duration and Transmission Method = FDD.

Note:	This operation must be performed after the Tx Gain Calibration Curve and the Tx DAC Calibration Curve calibrations.

Set up the equipment as follows:

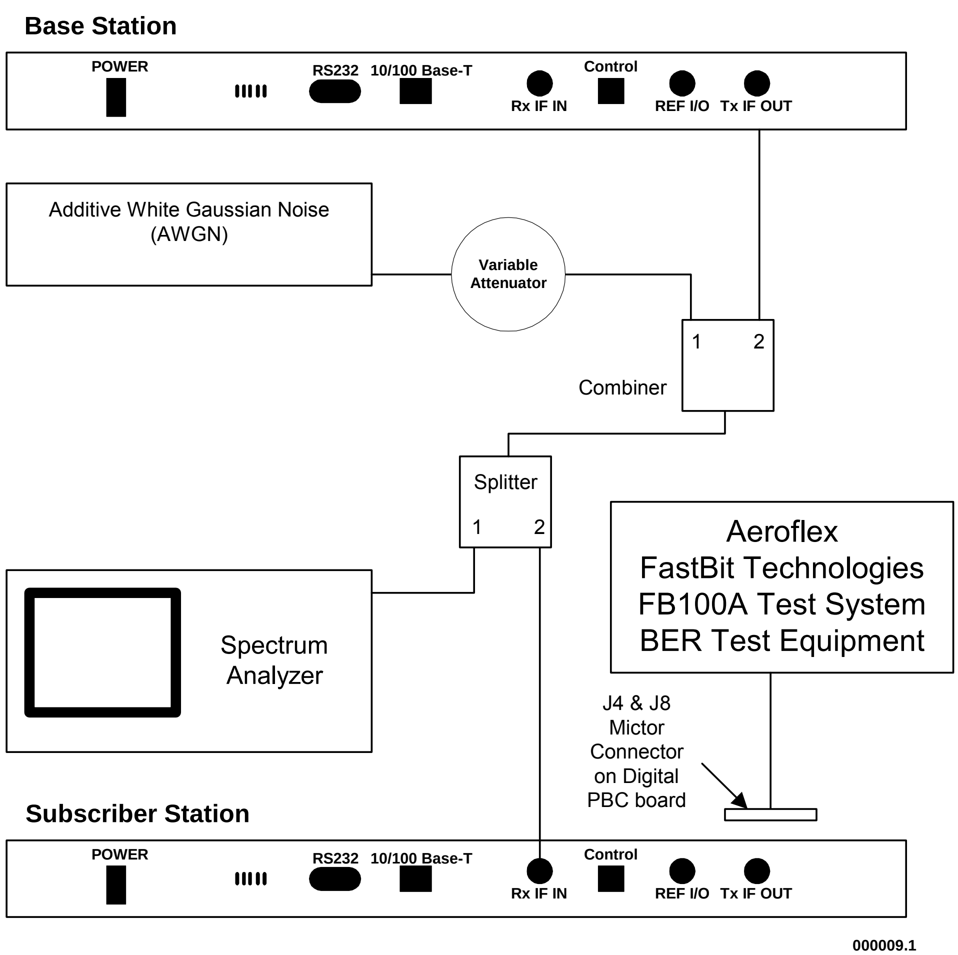


Figure 18 - Equipment Setup for Coded BER Test

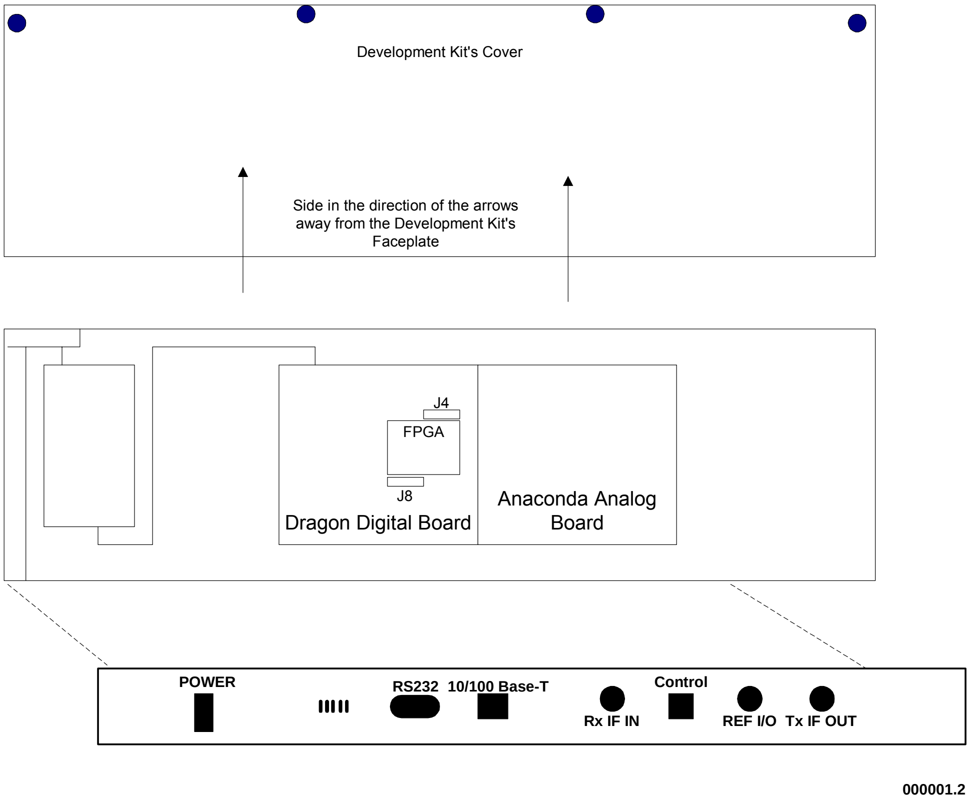


Figure 19 - Equipment Setup for Coded BER Test – J4 & J8 Male Mictor Connectors

1. To connect the BER Test equipment to the Subscriber Station's Development Kit,

   - Turn off the power to the Development Kit by toggling the on and off Power switch on the front panel of the Development Kit.
   - Remove the black top cover of the unit by un-screwing 4 Philip Head screws located to the back of the Development Kit.
   - The cover will slide away from the front face plate as you pull the cover in the opposite direction away from the face plate of the Development Kit.
   - To facilate BER Test a female Mictor Stake Adapter connector must be plugged to the J4 male Mictor connector which is located on top right side of the DM 256's FPGA which is located on the Digital Dragon Board.

- 

000040

Figure 20 - Sample Female Mictor Stake Adapter Connector

- Wavesat recommends the following Mictor Stake Adapter connector (MB002) by M6 Research, Inc. (www.m6research.com)

- You will need to connect the following pins on the Female Mictor at J4 to your BER Test Set.

- Wavesat recommends the following BERT Test Set: Aeroflex's Fast Bit Technologies FB100A Test System. www.aeroflex.com

Method 1:

Using two Female Mictor connectors on the Subscriber Station's Development Kit: Mictor connector 1 plugged at J4 and Mictor 2 connector plugged in at J8. This configuration will provide a clear clock from the Power PC.

- You will need to connect the following pins on the Female Mictor at J4 to your BER Test Set.

Mictor 1 Pin @ J4	Mictor Pin Description	BERT Analyzer	Description
13	DE12	D7	Data
21	DE8	GND or any GND	Ground
23	DE7	D valid	Data valid

- The second Mictor Stake Adapter connector must be used to connect to the J8 connector which is located on bottom left side of the DM 256's FPGA.

Mictor 2 Pin @ J8	Mictor Pin Description	BERT Analyzer	Description
5	CLKE	Clock	Clock

Turn on the power to the Development Kit by toggling on and off the Power switch and now follow the standard starting procedures as described earlier in the Development Guide Document to start up your Development Kit and continue testing.

Method 2:

Or alternative to the above configuration by using only one Female Mictor connector on Subscriber Station's Development Kit.

Mictor Pin	Mictor Pin Description	BERT Analyzer	Description
7	DE15	Clock	Clock
13	DE12	D7	Data
21	DE8	GND or any GND	Ground
23	DE7	D valid	Data valid

- Turn on the power to the Development Kit by toggling on and off the Power switch and now follow the standard starting procedures as described earlier in the Development Guide Document to start up your Development Kit and continue testing.

-Continue-

2. Set the base station development kit as follows:

- Configure the development kit by executing the following script on the Base Station, type:

  `./base ↵`

- To place the scheduler in a mode of operation to facilitate testing with a PRBS Analyzer on the Base Station, type:

  `./sched_ctrl set bert 1 ↵`

  `./phy_ctrl set tx_rand_stream 1 ↵`

- Setting up a connection by typing the following command:

`./sched_ctrl user modify 37 [DL_Modulation] [DL#_Allocation] QPSK-1/2 0 0 0 ↵`

Where

Parameter	Description
DL_Modulation	BPSK-1/2, BPSK-3/4, QPSK-1/2, QPSK-3/4, QAM16-1/2, QAM16-3/4, QAM64-2/3, QAM64-3/4
DL#_Allocation	The number of symbols in a frame for a specific DL Modulation excluding the 3 overhead symbols. (See Table below helps determine the maximum number of symbols in a frame for each specific Frame Duration Code).

Table 11 - # of Symbols/Frame with Different Frame Duration Codes in FDD Mode

Channel BW (MHz)	CP size	Frame Duration 2.5 (ms) for Frame Duration Code 0 #Symbols/ Frame	Frame Duration 4.00 (ms) for Frame Duration Code 1 #Symbols/ Frame	Frame Duration 5.00 (ms) for Frame Duration Code 2 #Symbols/ Frame	Frame Duration 8.00 (ms) for Frame Duration Code 3 #Symbols/ Frame	Frame Duration 10.00 (ms) for Frame Duration Code 4 #Symbols/ Frame	Frame Duration 12.50 (ms) for Frame Duration Code 5 #Symbols/ Frame	Frame Duration 20.00 (ms) for Frame Duration Code 6 #Symbols/ Frame
1.75	1/4	12.00	22.00	28.00	47.00	59.00	75.00	122.00
	1/8	14.00	24.00	31.00	52.00	66.00	83.00	135.00
	1/16	15.00	26.00	33.00	55.00	70.00	88.00	144.00
	1/32	15.00	27.00	34.00	57.00	72.00	91.00	148.00
3.5	1/4	28.00	47.00	59.00	97.00	122.00	153.00	247.00
	1/8	31.00	52.00	66.00	108.00	135.00	170.00	274.00
	1/16	33.00	55.00	70.00	114.00	144.00	180.00	291.00
	1/32	34.00	57.00	72.00	118.00	148.00	186.00	300.00
7	1/4	59.00	97.00	122.00	197.00	247.00	309.00	497.00
	1/8	66.00	108.00	135.00	219.00	274.00	344.00	552.00
	1/16	70.00	114.00	144.00	232.00	291.00	364.00	585.00
	1/32	72.00	118.00	148.00	239.00	300.00	375.00	603.00
10	1/4	86.00	139.00	175.00	282.00	354.00	443.00	711.00
	1/8	96.00	155.00	195.00	314.00	393.00	492.00	790.00
	1/16	102.00	165.00	207.00	333.00	417.00	522.00	837.00
	1/32	105.00	170.00	213.00	343.00	429.00	537.00	862.00

000039

Note for Channel BW 10 MHz:	When the D/A sampling clock for the DM256 is 40 MHz, the baseband sampling rate for the 10 Mhz channelization is Fs = 11.43 MHz, which is not the rate specified in the P802.16-2004 standard. To meet that requirement the D/A sampling (and A/D sampling clock) needs to be 40.32 MHz

-Continue-

3. Set the subscriber station development kit as follows:

   - Ensure that the following command has been run on the Subscriber Station prior to configurating the Subscriber Station script, type:

     ```
     ./phy_ctrl set rx_dma 0 ↵
     ```

   - Configure the development kit by executing the following script on the Subscriber Station, type:

     ```
     ./sub ↵
     ```

   - Configure the development kit serial port for testing, type:

     ```
     ./phy_ctrl set serial_port 1 ↵
     ```

   - Configure the development kit serial mode for testing, type:

     ```
     ./phy_ctrl set serial_mode 1 ↵
     ```

   - To obtain unbiased statistics, type:

     ```
     ./phy_ctrl set bad_data_filter 0 ↵
     ```

4. Set the noise level using the attenuator to where the SNR is high enough so the development kit is barely decoding any data, to allow the signal to be read and low enough to generate corrected errors.

   - To display the snr power results, type:

     ```
     ./phy_ctrl get snr_power ↵
     ```

-End-

10.8 SUI Channel Model Testing

Use this procedure to verify the development kit's performance in different environments by subjecting them to different Stanford University Interim (SUI) channel models. Six SUI models (1 to 6) and an additional one (LOS), which consists of an attenuated single only, can be used. This procedure can be performed using the following settings:

- CID: 10 (any CID between 1 and 36 can be used).
- Downlink symbols: 63; uplink symbols: 0 (to fill up the downlink sub-frame).
- This is dependant upon: Bandwidth = 3.5, CP Size = 1/16, Frame Duration = 2, Transmission method = FDD.

Note:	This operation must be performed after the Tx Gain Calibration Curve and the Tx DAC Calibration Curve calibrations.

Set up the equipment as shown below.

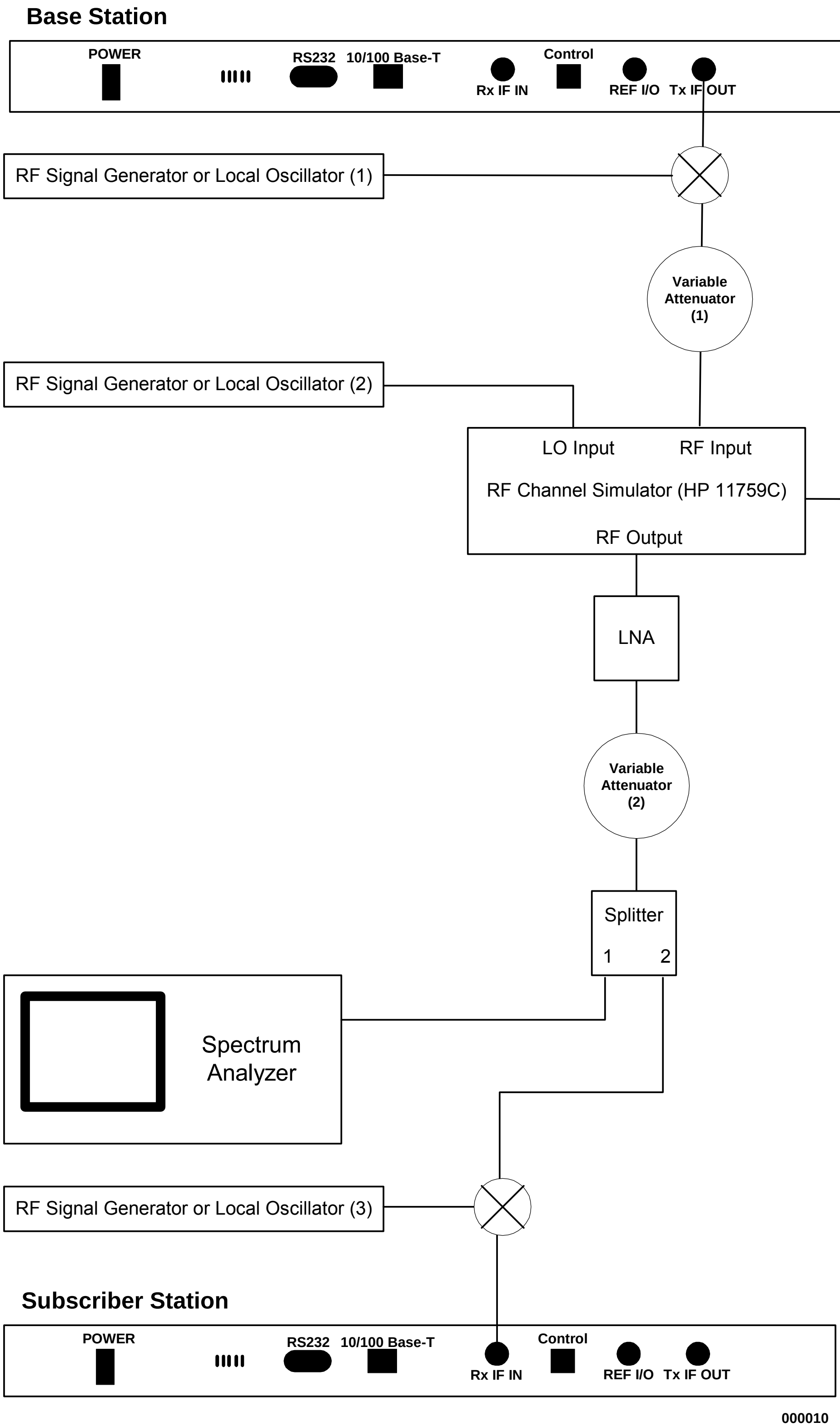


Figure 21 - Equipment Setup for SUI Test

1. Set the channel simulator to the desired SUI.
2. Set the base station development kit as follows:

   - Configure the development kit by executing the following script, type:

     ```
     ./base ↵
     ```

   - Add a connection by typing the following command:

   ```
   ./macdb_ctrl add 00:00:00:00:00:00 10 LinkTest QPSK-1/2 63 QPSK-1/2 0 0 0↵
   ```

   - Use the Spectrum Analyzer to verify the received signals.
3. Set the subscriber station development kit as follows:

   - Configure the development kit by executing the following script, type:

     ```
     ./sub ↵
     ```

4. Verify the received data and that you are receiving a SNR of around 30 by typing the following commands:

   ```
   ./ofdm_stats link 10 ↵
   ```

5. Set the RF generator (2) to 6 MHz below the RF input of the RF Channel Simulator.
6. Adjust Attenuator (1) as follows:

   - Without applying any load, turn on the RF generator (2).
   - If the “Power Over Range” light on the RF Channel Simulator turns On, lower the signal of the RF generator until it turns Off. If the light cannot be turned off, it means that the signal generated by the development kit is too strong, and must be attenuated using Attenuator (1).
   - If the “Power Over Range” light is not On, raise the signal of the RF generator until it comes On.
   - When you find the “Power Over Range” threshold, lower the RF generator about 2 dBm below the threshold.
7. Adjust Attenuator (2) as follows:

   - Ensure that Attenuator (2) is set to zero and look at the Spectrum Analyzer. A good signal should be around 20dBm
   - Verify that Rx power level is around 40 and that the SNR level is around 30 by typing the following command:

     ```
     ./ofdm_stats link 10 ↵
     ```

-Continue-

- If the signal is too weak, replace the Low Noise Amplifier (LNA) by another one with more gain.
- If the signal is too strong, reduce it using Attenuator (2).
-

8. Stop the transmission at the base station by executing the following command:

```
./sched_ctrl set pause 1 ↵
```

9. Reset the statistics at the subscriber station by typing the following command:

```
./ofdm_stats reset all ↵
```

10. Resume the transmission at the base station by typing the following command:

```
./sched_ctrl set pause 0 ↵
```

11. When you have run this experiment long enough (around 10 minutes), pause the transmission at the base station by typing the following command:

```
./sched_ctrl set pause 1 ↵
```

12. Record the number of good packets after executing the following command:

```
./ofdm_stats data 10 ↵
```

13. Record the number of lost packets after executing the following command:

```
./ofdm_stats ber 10 ↵
```

14. Calculate the packet loss rate using the following equation:

$$Y\% = \frac{lost\ \ packets}{lost\ \ packets + good\ \ packets} \times 100\%$$

15. Repeat this procedure for different profiles and SUIs.
16. Plot the packet loss rate against the SUI model used.

The standard deviation on the packet loss rate can be calculated using the following equation (from the derivative method of error estimation):

$$error = \frac{(lost\ \ packets) \times (good\ \ packets)^2 + (good\ \ packets) \times (lost\ \ packets)^2}{(Lost\ \ packets + good\ \ packets)^4} \times 100$$

Note:	To obtain lost packets as Bad Data in “data 10” the bad_data_filter must be set to 0.

-End-

10.9 Loopback Testing

The following loopback tests can be performed:

- Normal operating mode – no loopback.
- Raw data loopback (1).
- Packet loopback (2).
- Encoded word loopback (3).
- Sample feedback (4).
- IF loopback (5) – used for debug testing only.

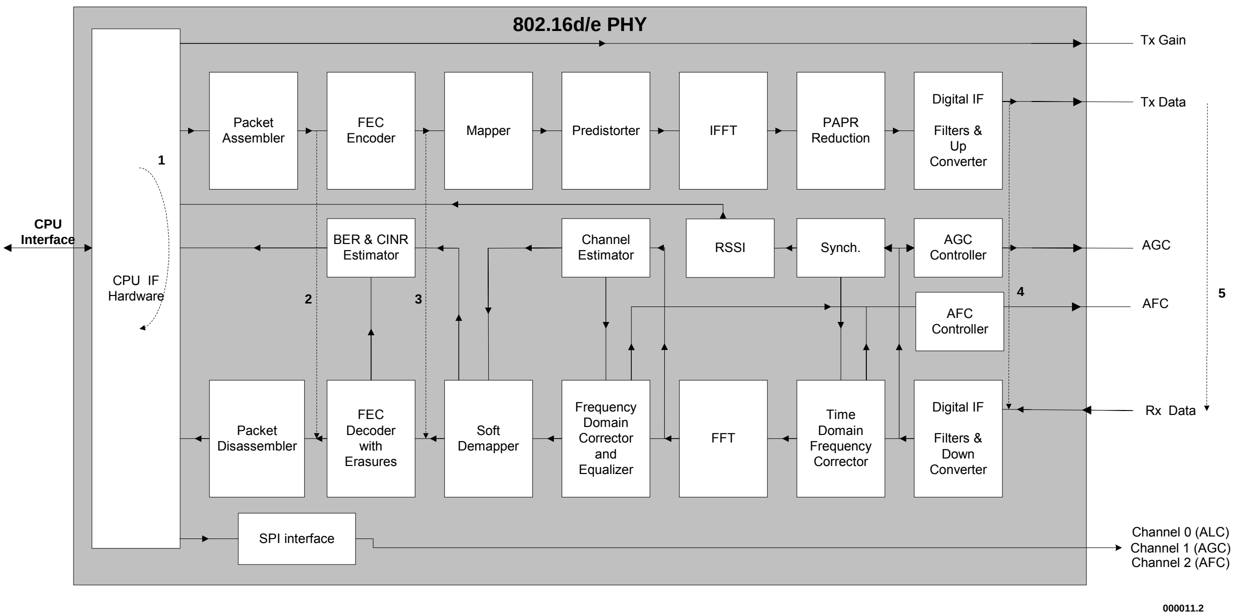


Figure 22 - Loopback Block Diagram Setup for SUI Test

Note:	This operation must be performed after the Tx Gain Calibration Curve and the Tx DAC Calibration Curve calibrations.

To configure DM256 for different types of loopback testing, type:

```
./sched_ctrl set feedback_mode value↵
```

Where

Parameter	Description
value	none (default) data (should not be used) packet encoder sample if

To show the loopback testing mode the DM256 has been set for, type:

```
./sched_ctrl get feedback_mode ↵
```

10.9.1 None Feedback Mode

This is the default mode, which means no DM 256 feedback mode is enabled and the scheduler operates normally, scheduling DL and UL traffic.

10.9.2 Data Feedback Mode

This mode should not be used.

10.9.3 Packet Feedback Mode

This mode places the DM 256 into packet feedback mode. The scheduler sets BS Mode = 1 and SS Mode = 0 and will only create and schedule transmit frames for DL. It will not schedule for the UL. As a result, only the first 4 burst described in the DLFP will be decoded.

10.9.4 Encoder Feedback Mode

This mode places the DM 256 into encoder feedback mode. The scheduler sets BS Mode = 1 and SS Mode = 0 and will only create and schedule transmit frames for DL. It will not schedule for the UL. As a result, only the first 4 burst described in the DLFP will be decoded.

10.9.5 Sample Feedback Mode

This mode places the DM 256 into sample feedback mode. The scheduler sets BS Mode = 1 and SS Mode = 0 and will only create and schedule transmit frames for DL. It will not schedule for the UL. As a result, only the first 4 burst described in the DLFP will be decoded.

10.9.6 IF Feedback Mode

This mode places the DM 256 into IF feedback mode. The scheduler sets BS Mode = 1 and SS Mode = 0 and will only create and schedule transmit frames for DL. It will not schedule for the UL. As a result, only the first 4 burst described in the DLFP will be decoded.

10.10 Advanced User Loopback Testing

Note:	The following loopback tests should only be performed by Advanced Users. The loopback test described below require the manual setting of other commands and parameters prior to testing.

The following loopback test are for advanced users to perform:

- Raw data loopback (1).
- Packet loopback (2).
- Encoded word loopback (3).
- Sample feedback (4).

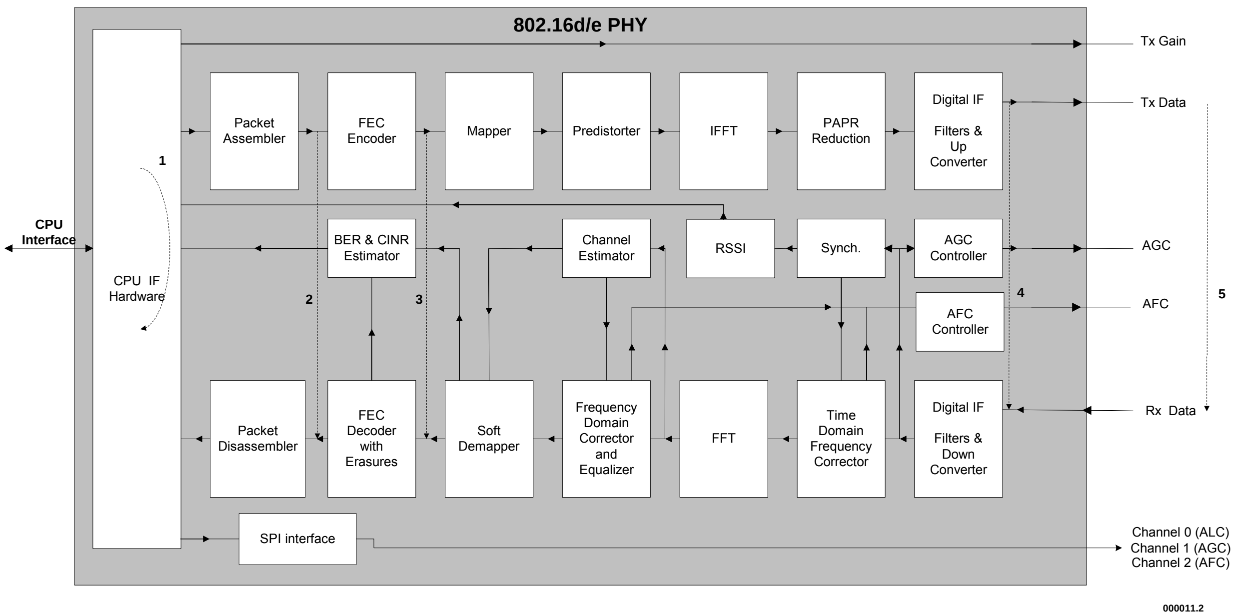


Figure 23 - Loopback Block Diagram Setup for SUI Test

Note:	This operation must be performed after the Tx Gain Calibration Curve and the Tx DAC Calibration Curve calibrations.

10.10.1 Data Loopback

To configure DM256 for data loopback testing, type:

```
./phy_ctrl set data_feedback value↵
```

Where

Parameter	Description
`value`	0= Disabled (normal setting) 1= Enabled

10.10.2 Packet Loopback

To configure DM256 for packet loopback testing, type:

```
./phy_ctrl set packet_feedback value↵
```

Where

Parameter	Description
value	0= Disabled (normal setting) 1= Enabled

10.10.3 Encoded Word Loopback

To configure DM256 for encoded word loopback testing, type:

```
./phy_ctrl set encode_feedback value↵
```

Where

Parameter	Description
value	0= Disabled (normal setting) 1= Enabled

10.10.4 Sample Loopback

To configure DM256 for sample loopback testing, type:

```
./phy_ctrl set sample_feedback value↵
```

Where

Parameter	Description
value	0= Disabled (normal setting) 1= Enabled

This page has been left intentionally blank.

Section: 11 General Configuration

This section provides the necessary information for users who wish to perform the following general configuration commands:

- To set the base station or a subscriber station in transmit mode.
- To set the base station or a subscriber station in receive mode.
- To set development kit's operating mode.
- To set the Syncronization of DL Frames to an external GPS 1 pulse/second synchronization signal.
- To interrupt the Tx data stream.
- To select the Packet Pass-Through mode.
- To set the Bad Data Filter so that the uncorrectable FEC blocks are discarded by the DM 256
- To set external reference clock.
- To set the antenna selection (this is not required for the development kit).
- To set the Base Station ID filter to compare the DM 256's Base ID valued contained in the DLFP and that of the sid register.
- To store the least significant 4 bits of the Base Station ID received in the DCD message.
- To set Synchronization Thresholds
-

Note:	The Enter key is noted by the following symbol: ↵

11.1 Transmit Mode

To set the development kit in Base Station mode or Subscriber Station mode for transmission, type:

```
./phy_ctrl set tx_bs_mode value↵
```

Where

Parameter	Description
[value]	0: subscriber station mode 1: base station mode

To show whether the development kit is in Base Station mode or Subscriber Station mode for transmission, type:

```
./phy_ctrl get tx_bs_mode ↵
```

11.2 Receive Mode

To set the development kit in Base Station mode or Subscriber Station mode for reception, type:

```
./phy_ctrl set rx_bs_mode value↵
```

Where

Parameter	Description
value	0: subscriber station mode 1: base station mode

To read the development kit in Base Station mode or Subscriber Station mode for reception, type:

```
./phy_ctrl get rx_bs_mode ↵
```

11.3 Operating Mode

Operating Mode	DM 256 Function during Operating Mode	Set By
0	DM 256 is configured for sample feedback.	By Default.
1	DM256 can transmit and receive data through the air interface	Can only be set after DM 256 has been fully configured.

To set the operating mode, type:

```
./phy_ctrl set operate_mode value ↵
```

Where

Parameter	Description
value	0: Standby/test mode 1: Normal operating mode

To read the operating mode, type:

```
./phy_ctrl get operate_mode ↵
```

11.4 GPS Slave Mode for Synchronization

This command is used in the Base Station mode to allow synchronization of the DL frames to an external GPS 1 pulse per second synchronization signal when enabled.

To enable the GPS slave mode, type:

```
./phy_ctrl set gps_slave value ↵
```

Where

Parameter	Description
value	0: disable GPS Slave Mode 1: enable GPS Slave mode

To display the GPS Slave mode, type:

```
./phy_ctrl get gps_slave ↵
```

11.5 Tx Enable Mode

The following command can be used to interrupt the Tx data stream before it leaves the chip.

To set the Tx enabled mode , type:

```
./phy_ctrl set tx_enable value↵
```

Where

Parameter	Description
value	0: Tx Enable mode is Off 1: Tx Enable mode is On (during normal operations, this function is always enabled.)

To show the whether the Tx enabled mode is on or off, type:

```
./phy_ctrl get tx_enable ↵
```

11.6 Packet Pass-Through Mode

To select the packet pass-through mode, type:

```
./phy_ctrl set pdu_passthrough value
```

Where

Parameter	Description
value	0: pass all packets to the Rx payload buffer 1: pass data packets only to the Rx payload buffer

To verify the packet pass-through mode, type:

```
./phy_ctrl get pdu_passthrough ↵
```

11.7 Bad Data Filter

This command is used to set bad_data filter on or off. When bad_data filter is set to on uncorrectable FEC blocks are discarded by the DM 256. When bad_data filter is set to off, uncorrectable FEC blocks are allowed to pass through. The only indication that they are not correctd is the Corrected Bit Error Count (RS) field in the pre-pendedheader, which will indicate 255.

To set the bad data filter, type:

```
./phy_ctrl set bad_data_filter value↵
```

Where

Parameter	Description
value	0: disable bad_data filter 1: enable bad_data filter

To verify the status of the bad_data filter, type:

```
./phy_ctrl get bad_data_filter ↵
```

11.8 External Reference Selection

Customers can use an external reference clock instead of the 10-MHz clock on the analog board. The external reference must be plugged into the REF I/O connector at the front of the development kit.

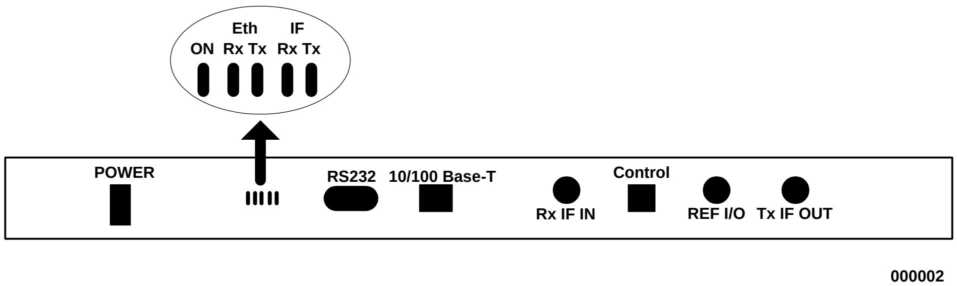


Figure 24 - Console Interface

To set the external reference, type:

```
./phy_ctrl set ext_ref value↵
```

Where

Parameter	Description
value	0: Use internal clock 1: Use external clock

To read the external reference, type:

```
./phy_ctrl get ext_ref ↵
```

11.9 Antenna Selection

Note:	This function is not required for the development kit.

It's a signal that allows customers to design transceivers that can switch between two antennas.

To switch between antennas, type:

```
./phy_ctrl set antenna_switch value ↵
```

Where

Parameter	Description
[value]	0: Antennas A is selected. 1: Antenna B is selected.

To read the transceiver selection, type:

```
./phy_ctrl get antenna_switch ↵
```

11.10 Base ID Enabled

These commands allow you to enable and disable the Base Station ID filetering by the DM 256. When Base Station ID filtering is enabled, the DM 256 will compare the Base Station ID value contained in the DLFP with the value contained in the base_id field in the sid register. When the Base Station ID filtering is disabled the DM 256 will not perform any filtering and will accept all frames.

To enable the Base Station ID filtering, type:

```
./phy_ctrl set base_id_enable value↵
```

Where

Parameter	Description
[value]	0: Disabled Base Station ID filtering. 1: enable Base Station ID filtering.

To display the status of the Base Station ID filtering, type:

```
./phy_ctrl get base_id_enable ↵
```

11.11 Base ID Register

These commands allow you to store and display the least significant 4 bits of the Base Station ID received in the DCD message. It is used by the DM 256 to filter frames for which the Base Sation ID field in the DLFP does not match the value contained in this register.

To store the least significant 4 bits of the Base Station ID received in the DCD message, type:

```
./phy_ctrl set base_id value ↵
```

Where

Parameter	Description
[value]	Base Station ID (0-15).

To display the current Base ID value of the sid register, type:

```
./phy_ctrl get base_id ↵
```

11.12 Synchronization Threshold

These commands will allow you to read and set the synchronization thresholds.

To set the REF1 Auto Correlation Threshold, type:

```
./phy_ctrl set ref1_auto_corr_thresh value↵
```

Where

Parameter	Description
value	Recommended value 125

To read REF1 Auto Correlation Threshold value, type:

```
./phy_ctrl get ref1_auto_corr_thresh ↵
```

To set the REF2 Auto Correlation Threshold, type:

```
./phy_ctrl set ref2_auto_corr_thresh value↵
```

Where

Parameter	Description
value	Recommended value 188

To read REF2 Auto Correlation Threshold value, type:

```
./phy_ctrl get ref2_auto_corr_thresh ↵
```

To set the REF2 Cross Correlation Threshold, type:

```
./phy_ctrl set ref2_cross_corr_thresh value↵
```

Where

Parameter	Description
value	Recommended value 63

To read REF2 Cross Correlation Threshold value, type:

```
./phy_ctrl get ref2_cross_corr_thresh ↵
```

This page has been left intentionally blank.

Section: 12 Gain Control

This section provides the necessary information for users who wish to perform the following:

- To control the Peak to Average Power Ratio
- To modify the Tx digital Gain scale value.

Note:	The Enter key is noted by the following symbol: ↵

12.1 Peak-to-Average Power Ratio (PAPR) Control

The Peak-to-Average Power Ratio (PAPR) can be controlled using the following parameters:

- Tx scaling factor.
- Rx scaling factor.

12.1.1 Tx Scaling Factor

To set the Tx scaling, type:

```
./phy_ctrl set tx_scaling value↵
```

Where

Parameter	Description
`value`	4-bit 2's complement value ranging from decimal (-8 to +7). See table below. (recommended value: -1)

Note:	A value of -1 will prevent the chip from being saturated at any Tx gain value. A value of 0 will provide a stronger signal, which however can be saturated and reduce the PAPR at some Tx gain values. The overall controllable PAPR range is in the order of 3-4 db.

Unsigned	15	14	13	12	11	10	9	8
Signed	-1	-2	-3	-4	-5	-6	-7	-8
Unsigned	7	6	5	4	3	2	1	0
Signed	7	6	5	4	3	2	1	0

To show the Tx scaling which is a 4-bit unsigned value, type:

```
./phy_ctrl get tx_scaling ↵
```

12.1.2 Rx Scaling Factor

To set the Rx scaling, type:

```
./phy_ctrl set rx_scaling value↵
```

Where

Parameter	Description
value	4-bit 2's complement value ranging from decimal (-8 to +7). See table below. (recommended value: 0)

Unsigned	15	14	13	12	11	10	9	8
Signed	-1	-2	-3	-4	-5	-6	-7	-8
Unsigned	7	6	5	4	3	2	1	0
Signed	7	6	5	4	3	2	1	0

To show the Rx scaling, type:

```
./phy_ctrl get rx_scaling ↵
```

12.2 Tx Digital Gain

The Tx digital gain can be modified by:

- Setting (specifying) a scale value.
- Forcing (applying) the scale value.

12.2.1 Setting the Tx Gain Value

The Tx digital gain can be modified by specifying a scale value between 0 and 255. By default, the value is at 250.

To set the Tx gain value, type:

```
./phy_ctrl set tx_gain_value value↵
```

Where

Parameter	Description
[value]	0 – 255 (The default value is 250)

To show the Tx gain value, type:

```
./phy_ctrl get tx_gain_value ↵
```

12.2.2 Forcing the Tx Gain

After setting (specifying) a Tx gain value, the setting must be forced (applied). If the Tx gain value is not forced, then, the Tx gain value of 255 (maximum gain) is used by default.

To force the Tx gain, type:

```
./phy_ctrl set tx_force_gain value↵
```

Where

Parameter	Description
[value]	0: Use default Tx gain factor (255) 1: Apply user-defined Tx gain factor

To show the Tx gain, type:

```
./phy_ctrl get tx_force_gain ↵
```

This page has been left intentionally blank.

Section: 13 AFC, AGC, ALC Configuration

DM256 provides Automatic Frequency Correction (AFC) to compensate for any discrepancy in frequency between a Subscriber Station and the Base Station. The AFC is connected to the Voltage Control (VC) pin of the 10MHz VCTCXO (Voltage Controlled Temperature Compensated Crystal Oscillator).

DM256 provides Automatic Level Control and Automatic Gain Control to compensate for the signal loss caused by distance separation between a Subscriber Station and the Base Station. Both ALC and AGC are performed at the Subscriber Station.

	ASIC / FPGA When external DACs are used via the SPI interface				ASIC When built in DACs are used			
	UL		DL		UL		DL	
	BS rx_bs_1_ss_0 = 1	SS tx_bs_1_ss_0 = 0	BS tx_bs_1_ss_0 = 1	SS rx_bs_1_ss_0 = 0	BS rx_bs_1_ss_0 = 1	SS tx_bs_1_ss_0 = 0	BS tx_bs_1_ss_0 = 1	SS rx_bs_1_ss_0 = 0
AFC				√ (2)				√ (2)
AGC	√ (1) special test			√ (2)				√ (2) √ (3)
ALC		√ (2)				√ (2)		

(1): For test purpose only (should not be enabled in regular operating mode). Value will be refreshed with AGC unit value when a good OFDM symbol is received. (When gets an “rx_ofdm_frame” from the “synchronizer”)

(2) : Debug mode is working properly only with external DACs via SPI interface. Forced will be kept when in “debug mode”.

Debug mode is not working properly when used with internal DAC’s. The forced value will be overriden by AFC, AGC or ALC units.

(3): Response has one frame delay added compares to SPI external DAC.

13.1 AFC DAC Register

To obtain the contents of the AFC DAC register, type:

```
./phy_ctrl get afc ↵
```

13.2 AFC Module

To enable the AFC module type:

```
./phy_ctrl set afc_enable value↵
```

Where

Parameter	Description
value	0= disable the AFC Module 1= enable the AFC Module.

To show the status of the AFC Module, type:

```
./phy_ctrl get afc_enable ↵
```

13.3 AGC Module

To enable the AGC module type:

```
./phy_ctrl set agc_enable value↵
```

Where

Parameter	Description
value	0= disable the AGC Module 1= enable the AGC Module.

To show the status of the AGC Module, type:

```
./phy_ctrl get agc_enable ↵
```

13.4 AGC DAC Register

To obtain the contents of the AGC DAC register, type:

```
./phy_ctrl get agc ↵
```

13.5 VGA

To set a constant value for VGA in the AGC module,type:

```
./phy_ctrl set vga value↵
```

Where

Parameter	Description
value	Value of the VGA gain of the amplifier used in the Rx Path 50 dB = Decimal 81 = (000001010001)

To show the value of the VGA algorithm in the AGC Module, type:

```
./phy_ctrl get vga↵
```

Note:	The value shown is the inverse of the actual value.

13.6 ALC DAC Register

To set a constant value for the ALC module type:

```
./phy_ctrl set alc value↵
```

Where

Parameter	Description
value	Value of the ALC DAC Register (0-1024)

13.7 ALC Module

To enable the ALC module type:

```
./phy_ctrl set alc_enable value↵
```

Where

Parameter	Description
value	0= disable the ALC Module 1= enable the ALC Module.

To show the status of the ALC Module, type:

```
./phy_ctrl get alc_enable ↵
```

This page has been left intentionally blank.

Section: 14 Digital-to-Analog Converter Configuration

DM256 uses digital-to-analog converters (DACs) to implement the following functions:

- Automatic frequency correction (AFC) via the Phased Lock Loop (PLL) DAC.
- Automatic level control (ALC) via the Tx DAC.
- Automatic gain control (AGC) via the Rx DAC.

The DACs reside on the analog board and are connected to DM256 via the Serial Peripheral Interface (SPI). The automatic functions (mentioned above) are operational when the SPI is placed in automatic mode. In this mode, the DACs are automatically set by DM256.

Note:	The SPI must be set to manual mode before all other commands are used.

When the SPI is placed in manual mode, the DACs must be set manually by the user.

To use the DACs in manual mode, the following operations are necessary:

1. Set the SPI to manual mode.
2. Enable the SPI.
3. Set the SPI clock, if necessary.
4. Apply the user-defined SPI clock setting, if necessary.
5. Set the PLL DAC manually.
6. Set the Tx DAC manually.
7. Set the Rx DAC manually

Note:	The Enter key is noted by the following symbol: ↵

14.1 Setting the SPI to Manual Mode

To apply the values set for the DACs, the latter must be placed in manual mode.

To set the SPI to Manual mode, type:

```
./phy_ctrl set spi_manual_mode_enable value↵
```

Where

Parameter	Description
value	0= automatic mode 1= manual mode

To show the SPI is set to manual or automatic mode, type:

```
./phy_ctrl get spi_manual_mode_enable ↵
```

14.2 Enabling the SPI

DM256 controls the DACs on the analog board via the SPI. To use the DACs, the SPI must be enabled.

To enable or disable the SPI, type:

```
./phy_ctrl set spi_enable value↵
```

where

Parameter	Description
value	0: disable SPI 1: enable SPI (normal setting)

To display the SPI Interface, type:

```
./phy_ctrl get spi_enable ↵
```

14.3 Setting the SPI Clock

The following command allows the user to set the SPI clock speed. The user-defined setting is used only if the SPI clock is enabled. If the SPI clock is disabled, the default setting is used (6.250 MHz).

To set the SPI clock speed, type:

```
./phy_ctrl set spi_clock_speed value↵
```

Where

Parameter	Description
value	0 = 6.250 MHz 1 = 3.125 MHz 2 = 1.562 MHz 3 = 781.250 kHz 4 = 390.625 kHz 5 = 195.312 kHz 6 = 97.656 kHz 7 = 48.828 kHz 8 = 24.414 kHz 9 = 12.207 kHz 10 –15 = 6.250 MHz

Note:	Recommended setting: 0 (changing this value will slow down the SPI processing.

To display the SPI clock speed, type:

```
./phy_ctrl get spi_clock_speed ↵
```

14.4 Applying the Clock Setting

The following command is used to enable the user-defined setting of the clock. If the user-defined setting is not enabled, the SPI clock uses its default setting (6.250 MHz).

To enable the SPI Clock, type:

```
./phy_ctrl set spi_clock_enable value↵
```

Parameter	Description
value	0: Set to default value (6.250 MHz) 1: Set to user-defined value (see above)

To display the SPI Clock setting, type:

```
./phy_ctrl get spi_clock_enable ↵
```

14.5 Setting the PLL DAC manually

The following command is used to set the PLL DAC to a scale value between 0 and 255. That setting is used only if the SPI is set to manual mode. If the SPI is in automatic mode, the PLL DAC is set via the automatic frequency control (AFC).

To set the AFC DAC, type:

```
./phy_ctrl set spi_pll_dac value↵
```

where

Parameter	Description
value	0…255

14.6 Setting the Tx DAC manually

The following command is used to set the Tx DAC to a scale value between 0 and 255. That setting is used only if the SPI is set to manual mode. If the SPI is in automatic mode, the Tx DAC is set via the automatic level control (ALC).

To set the Tx DAC, type:

```
./phy_ctrl set spi_tx_dac value↵
```

where

Parameter	Description
value	0…255

14.7 Setting the Rx DAC manually

The following command is used to set the Rx DAC to a scale value between 0 and 255. That setting is used only if the SPI is set to manual mode. If the SPI is in automatic mode, the Rx DAC is set via the automatic gain control (AGC).

To set the Rx DAC, type:

```
./phy_ctrl set spi_rx_dac value↵
```

where

Parameter	Description
value	0…255

Section: 15 Analog Control and Analog Value Register

The following commands are used for Analog testing and Analog Configuration purposes.

Note:	The Enter key is noted by the following symbol: ↲

15.1 Setting Analog Test

To set the analog test mode, type:

```
./phy_ctrl set analog_test value↲
```

Where

Parameter	Description
value	0= disable mode, analog test: tx sample DAC come from the Tx Digital IF, AFC DAC = AFC module, AGC DAC = AGC module, ALC DAC = ALC in the bus interface 1= enable mode, analog test: tx samples DAc, AFC DAC, AGC DAC, and ALC DAC values all come from the Analog register in the bus interface.

To show the analog test mode, type:

```
./phy_ctrl get analog_test ↲
```

15.2 Obtain values for inphase ADC

To show the values for inphase ADC, type:

```
./phy_ctrl get analog_adc_inphase ↲
```

Note:	Development Kit must be in debug mode before using this command, i.e. ./phy_ctrl set analog_test 1 ↲

15.3 Setting Analog ADC Module

To set the Analog ADC module, type:

```
./phy_ctrl set analog_adc_mode value↵
```

Where

Parameter	Description
Value	0 = Digital mode, Rx input are read from 10-bit RX_SAMPLES_PIN bus, Power Down ADC. 1= Analog mode, Internal ADC Standby. 2= Analog mode, Internal ADC using unsigned binary coding. 3 = Analog mode, Internal ADC using 2's complement coding.

To show the Analog ADC Modules mode, type:

```
./phy_ctrl get analog_adc_mode ↵
```

15.4 Obtain values for quadrature ADC

To show the values for quadrature ADC, type:

```
./phy_ctrl get analog_quad ↵
```

Note:	Development Kit must be in debug mode before using this command, i.e. ./phy_ctrl set analog_test 1 ↵

15.5 Setting Analog AFC Module

To set the AFC DAC module, type:

```
./phy_ctrl set analog_afc value↵
```

Where

Parameter	Description
Value	0= disable AFC DAC Module 1= enable AFC DAC Module

To show the AFC DAC Modules mode, type:

```
./phy_ctrl get analog_afc ↵
```

15.6 Setting Analog ALC Module

To set the ALC DAC module, type:

```
./phy_ctrl set analog_alc value↵
```

Where

Parameter	Description
Value	0= disable ALC DAC Module 1= enable ALC DAC Module

To show the ALC DAC Modules mode, type:

```
./phy_ctrl get analog_alc ↵
```

15.7 Setting Analog DAC

To set all analog DAC, type:

```
./phy_ctrl set analog_dac value↵
```

Where

Parameter	Description
Value	Set value for all Analog DACs when in Analog Test Mode.

To show the value of all analog DACs, type:

```
./phy_ctrl get analog_dac ↵
```

15.8 Setting Analog Test Pulse

To set the analog Test Pulse, type:

```
./phy_ctrl set analog_test_pulse value↵
```

Where

Parameter	Description
value	0= disable mode 1= enable mode

To show the analog Test Pulse mode, type:

```
./phy_ctrl get analog_test_pulse ↵
```

15.9 Setting Analog Tx DAC inphase DAC

To set the analog Tx DAC inphase DAC, type:

```
./phy_ctrl set analog_tx_inphase value↵
```

Where

Parameter	Description
value	0= disable mode 1= enable mode

To show analog Tx DAC inphase DAC mode, type:

```
./phy_ctrl get analog_tx_inphase ↵
```

15.10 Setting Analog Tx Sample pins

To set the 10 bits digital Tx sample pins, type:

```
./phy_ctrl set analog_tx_samples_disable value↵
```

Where

Parameter	Description
value	0= disable mode, driven by the Tx Digital IF 1= enable mode, driven to 0

To show 10 bits digital Tx sample pins driven by Tx Digital IF or driven to 0, type:

```
./phy_ctrl get analog_tx_samples_disable ↵
```

15.11 Setting Tx DAC quadrature DAC

To set the Tx DAC quadrature DAC, type:

```
./phy_ctrl set analog_tx_quad value↵
```

Where

Parameter	Description
value	0= disable mode 1= enable mode

To show Tx DAC quadrature DAC mode, type:

```
./phy_ctrl get analog_tx_quad ↵
```

15.12 Setting Sleep Mode in DAC

To set the DAC to sleep mode, type:

```
./phy_ctrl set analog_tx_sleep value↵
```

Where

Parameter	Description
value	0= disable mode 1= enable mode

To show DAC sleep mode is enabled or not, type:

```
./phy_ctrl get analog_tx_sleep ↵
```

This page has been left intentionally blank.

Section: 16 IF Configuration

By default, the Tx center frequency and Rx center frequency, must be set to 1/4 of sampling clock and their bandwidths are set at 3.5 MHz.

The following commands and parameters can be used to modify these settings:

- IF Digital to Analog Converter and Analog to Digital Converter Interface.
  - Rx Analog to Digital Converter Interface.
  - Tx Digital to Analog Converter Interface.
- Sampling clock/frequency.
- Tx Frequency Ratio.
- Tx Center Frequency.
- Rx Frequency Ratio.
- Rx Center Frequency.
- Tx and Rx Channel Bandwidth at base station and subscriber station.
- User specified Tx and Rx Center Frequency and Bandwidth.
- Frequency Output Type.

Note:	The Enter key is noted by the following symbol: ↵

16.1 IF Digital to Analog Converter and Analog to Digital Converter Interface

16.1.1 Rx Analog to Digital Converter Interface (Commands for ASIC Use Only)

The following procedure will allow you to set the Rx ADC digital bus mode to 2’s complement or straight bit.

To set the Rx ADC digital bus mode, type:

```
./phy_ctrl set rx_adc_mode value ↵
```

Where

Parameter	Description
value	0: enable straight bit format (for built in ADCs) 1: enable 2’s bit complement (default for built in ADCs).

To determine the Rx ADC digital bus mode, type:

```
./phy_ctrl get rx_adc_mode ↵
```

16.1.2 Tx Digital to Analog Converter Interface (Commands for ASIC Use Only)

The following procedure will allow you to set the Tx DAC digital bus mode to 2's complement or straight bit.

To set the Tx DAC digital bus mode, type:

```
./phy_ctrl set tx_dac_mode value ↵
```

Where

Parameter	Description
`value`	0: Disable DAC. 1: enable 2's bit complement (default for built in DACs).

To determine the Tx DAC digital bus mode, type:

```
./phy_ctrl get tx_dac_mode ↵
```

16.2 Setting the Sampling Clock

This procedure is used to control the PLL (Phase Lock Loop) with its reference input pin connected to the 10 MHz Voltage Controlled Temperature Compensated Crystal Oscillator (VCTCXO) on the analog board. The frequency outputted by the clock is fed to DM256 for Rx and Tx signal sampling.

To set the sampling frequency, type:

```
./anaconda_ctrl set sample_clk value↵
```

Where

Parameter	Description
`value`	Sampling frequency in Hz. Possible values are: - 13000000 (13.0 MHz) - 33333333 (33.3 MHz) - 40000000 (40.0 MHz) - 66666666 (66.6 MHz)

Note:	The setting that must be used for the development kit is 40000000.

To read the sampling clock frequency on the analog board, type:

```
./anaconda_ctrl get sample_clk ↵
```

16.3 Setting the Tx Center Frequency

The Tx frequency must be set both at the base station and at the subscriber station using either of the following parameters:

- Tx Frequency Ratio. (Please see DM 256 Data Book (256-59-01) Section on Register Descriptions sub section Digital IF Register 1 and 2, for more information concerning this ratio)
- Tx Center Frequency.

Note:	There is no limitation on the value specified for the Rx center frequency. For example, if the Rx center frequency is chosen to be –5 MHz, then the inverted image at –5 MHz will be demodulated.

16.3.1 Tx Frequency Ratio

The Tx Frequency Ratio is used to set the Tx center frequency as a ratio of the clock sampling frequency.

For example:

To set the Tx center frequency to 5 MHz based on a 40-MHz clock, multiply 0.125 [which is 5 MHz divided by 40 MHz sampling clock] by 65536 [16-bit number] = 0.125 * 65536 = 8192. However, the Tx Frequency Ratio can only be a multiple of 16. Therefore, 8192/16 = 512 is a valid Tx Frequency Ratio step.

To set the Tx Frequency Ratio, type:

```
./phy_ctrl set tx_freq_ratio value↵
```

Where

Parameter	Description
value	16384

To read the Tx Frequency Ratio, type:

```
./phy_ctrl get tx_freq_ratio ↵
```

16.3.2 Tx Center Frequency

The Tx center frequency is calculated as a ratio of the clock sampling frequency. Therefore, when you specify the Tx center frequency using the following command, the actual value that will be used is a ratio of the clock sampling frequency that is closest to the specified value. For that reason, the Tx center frequency that is returned by the console might not be exactly the value that you have specified.

For example:

If you set the Tx center frequency to 3500000, the actual value that will be used is 3498831.

To set the Tx Center Frequency, type:

```
./phy_ctrl set tx_center_freq value↵
```

Where

Parameter	Description
value	16384

To read the Tx Center Frequency, type:

```
./phy_ctrl get tx_center_freq ↵
```

16.4 Setting the Rx Center Frequency

The Rx frequency must be set both at the base station and at the subscriber station using either of the following parameters:

- Rx Frequency Ratio. (Please see DM 256 Data Book (256-59-01) Section on Register Descriptions sub section Digital IF Register 1 and 2, for more information concerning this ratio)
- Rx Center Frequency.

Note:	There is no limitation on the value specified for the Rx center frequency. For example, if the Rx center frequency is chosen to be –5 MHz, then the inverted image at –5 MHz will be demodulated.

16.4.1 Rx Frequency Ratio

The Rx Frequency Ratio is used to set the Rx center frequency as a ratio of the clock sampling frequency.

For example:

To set the Rx center frequency to 5 MHz based on a 40-MHz clock, multiply 0.125 [which is 5 MHz divided by 40 MHz sampling clock] by 65536 [16-bit number] = 0.125 * 65536 = 8192. . However, the Tx Frequency Ratio can only be a multiple of 16. Therefore, 8192/16 = 512 is a valid Tx Frequency Ratio steps.

To set the Rx Frequency Ratio, type:

```
./phy_ctrl set rx_freq_ratio value↵
```

Where

Parameter	Description
value	16384

To read the Rx Frequency Ratio, type:

```
./phy_ctrl get rx_freq_ratio ↵
```

16.4.2 Rx Center Frequency

The Rx center frequency is calculated as a ratio of the clock sampling frequency. Therefore, when you specify the Rx center frequency using the following command, the actual value that will be used is a ratio of the clock sampling frequency that is closest to the specified value. For that reason, the Rx center frequency that is returned by the console might not be exactly the value that you have specified.

For example:

If you set the Rx center frequency to 3500000, the actual value that will be used is 3498831.

To set the Rx Center Frequency, type:

```
./phy_ctrl set rx_center_freq value ↵
```

Where

Parameter	Description
value	16384

To read the Rx Center Frequency, type:

```
./phy_ctrl get rx_center_freq ↵
```

16.5 Setting the Channel Bandwidth

By default, the channel bandwidth is at 3.5 MHz. This parameter can be modified using commands. The commands for setting the channel bandwidth are different between the base station and the subscriber station.

16.5.1 Tx and Rx Bandwidth (at the Base Station)

Note:	The channel bandwidth cannot be modified once the scheduler is started; i.e. once the “./sched_ctrl set start” command has been executed (set to 0). Therefore if you wish to modify this parameter, you need to reboot the development kit first, change the parameter setting in the base station script and then apply the script. Ensure that the “./sched_ctrl set start” is the last command in the script..

If you intend to modify the Tx and Rx center frequency at the base station, the modifications must be performed before using the following command in order to apply them.

To set the channel bandwidth (Tx and Rx) at the base station, type:

```
./sched_ctrl set channel_bw value↵
```

Where

Parameter	Description
value	1.75, 3.5, 7, 10

To read the channel bandwidth, type:

```
./sched_ctrl get channel_bw ↵
```

16.5.2 Tx Bandwidth (at the Subscriber Station)

To set the Tx bandwidth at the subscriber station, type:

```
./phy_ctrl set tx_bw_mode value↲
```

Where

Parameter	Description
value	1.75, 3.5, 7, 10

To read the Tx bandwidth value, type:

```
./phy_ctrl get tx_bw_mode ↲
```

16.5.3 Rx Bandwidth (at the Subscriber Station)

To set the Rx bandwidth at the subscriber station, type:

```
./phy_ctrl set rx_bw_mode value↲
```

Where

Parameter	Description
value	1.75, 3.5, 7, 10

To read the Rx bandwidth value, type:

```
./phy_ctrl get rx_bw_mode ↲
```

16.6 Applying the Center Frequencies and Bandwidth

By default, the Tx Center Frequency and the Rx Center Frequency must be both set at 1/4 of clock sampling frequency and the bandwidths are set at 3.5 MHz. If you have specified different center frequencies or bandwidths, you must apply them using the commands described in the following subsections. These commands are necessary only at the subscriber station.

16.6.1 Tx Center Frequency

Note:	This command is for Wavesat Use Only.

To create a dc carrier at the set center frequency at the subscriber station, type:

```
./phy_ctrl set tx_cont_wave value↵
```

Where

Parameter	Description
value	0 = normal operation mode. 1 = in test mode for analog side, sin wave at Center Frequency set according to./phy_ctrl set tx_freq_ratio command.

To show the specified Tx Center Frequency at the subscriber station, type:

```
./phy_ctrl get tx_cont_wave ↵
```

16.6.2 Tx Bandwidth and Center Frequency

Note:	The following command ./phy_ctrl set digital_if_force_tx 1 ↵ must be invoked in order to activate the previous commands such as Tx Bandwidth and Center Frequency.

To force the specified Tx bandwidth and Center Frequency at the subscriber station, type:

```
./phy_ctrl set digital_if_force_tx value↵
```

Where

Parameter	Description
value	0 = Use default values 1 = Use specified Tx bandwidth and Center Frequency

To show the specified Tx bandwidth and Center Frequency at the subscriber station, type:

```
./phy_ctrl get digital_if_force_tx ↵
```

16.6.3 Forcing the Specified Rx Bandwidth and Center Frequency

Note:	The following command ./phy_ctrl set digital_if_force_rx 1 ↵ must be invoked in order to activate the previous commands such as Rx Bandwidth and Center Frequency.

To force the specified Rx bandwidth and center frequency at the subscriber station, type:

```
./phy_ctrl set digital_if_force_rx value↵
```

Where

Parameter	Description
value	0 = Use default values 1 = Use specified Rx bandwidth and Center Frequency

To show the specified Rx bandwidth and center frequency at the subscriber station, type:

```
./phy_ctrl get digital_if_force_rx ↵
```

16.7 Selecting the Frequency Output Type

16.7.1 Selecting Tx Frequency output I/Q and IF (ASIC command only)

To select between I/Q and IF output for Tx, type:

```
./phy_ctrl set tx_digital_iq_output value↵
```

Where

Parameter	Description
value	0 = IF output mode (by default) 1 = I/Q output mode

To show the Tx Frequency output type:

```
./phy_ctrl get tx_digital_iq_output ↵
```

16.7.2 Selecting Rx Frequency output I/Q and IF (ASIC command only)

To select between I/Q and IF output for Rx, type:

```
./phy_ctrl set rx_digital_iq_output value↵
```

Where

Parameter	Description
value	0 = IF output mode (by default) 1 = I/Q output mode

To read the Rx Frequency output I/Q or IF, type:

```
./phy_ctrl get rx_digital_iq_output ↵
```

Section: 17 Channel Coding

This section provides the necessary information for users who wish to set the following:

- Tx interleaver.
- Rx deinterleaver.
- Tx randomizer.
- Rx derandomizer.

Note:	The Enter key is noted by the following symbol: ↵

17.1 Tx Interleaver

Note:	Tx interleaver and Rx deinterleaver work together; i.e. when one is enabled, the other must also be enabled and vice versa. It is recommended that both functions be enabled otherwise the effectiveness of the transmission is reduced.

To set the Tx Interleaver, type:

```
./phy_ctrl set tx_interleaver value ↵
```

Where

Parameter	Description
value	0: enable 1: disable

To verify the Tx Interleaver, type:

```
./phy_ctrl get tx_interleaver ↵
```

17.2 Rx Deinterleaver

Note:	Tx interleaver and Rx deinterleaver work together; i.e. when one is enabled, the other must also be enabled and vice versa. It is recommended that both functions be enabled otherwise the effectiveness of the transmission is reduced.

To set the Rx deinterleaver, type:

```
./phy_ctrl set rx_deinterleaver value ↵
```

Where

Parameter	Description
value	0: enable 1: disable

To verify the Rx deinterleaver, type:

```
./phy_ctrl get rx_deinterleaver ↵
```

17.3 Tx Randomizer

Note:	Tx Randomizer and Rx Derandomizer work together; i.e. when one is enabled, the other must also be enabled and vice versa. It is recommended not to disable these functions.

To set the Tx randomizer, type:

```
./phy_ctrl set tx_randomizer value ↵
```

Where

Parameter	Description
value	0: enable 1: disable

To verify the Tx randomizer, type:

```
./phy_ctrl get tx_randomizer ↵
```

17.4 Rx Derandomizer

Note:	Tx Randomizer and Rx Derandomizer work together; i.e. when one is enabled, the other must also be enabled and vice versa. It is recommended not to disable these functions.

To set the Rx derandomizer, type:

```
./phy_ctrl set rx_derandomizer value ↵
```

Where

Parameter	Description
value	0: enable 1: disable

To verify the Rx derandomizer, type:

```
./phy_ctrl get rx_derandomizer ↵
```

This page has been left intentionally blank.

Section: 18 Delay Compensation

This section provides the necessary information for users who wish to set or perform the following:

- Delay correction on subscriber development kit.
- Auto Tx Mute delay lead.
- Auto Tx Mute delay lag.
- Disable the Auto Tx mute.
- Control Tx Mute manually.

Note:	The Enter key is noted by the following symbol: ↵

18.1 Delay Correction

The delay correction can be set at the subscriber development kit only and is used to compensate for the time delay between when a burst is scheduled for transmission and when it is received at the base station development kit. This parameter must be set such that the range delay that is read at the base station (using the ./ofdm_stats link) reads 0 (± 1). This parameter is used during initial ranging and periodic ranging.

To set the delay correction, type:

```
./phy_ctrl set delay_correction value ↵
```

Where

Parameter	Description
value	16-bit unsigned number of sample periods to advance transmission.

To display the number of complex samples set for Tx delay correction, type:

```
./phy_ctrl get delay_correction ↵
```

18.2 Auto Tx Mute

The Tx mute function can be controlled as follows:

- Setting the auto Tx mute delay lead.
- Setting the auto Tx mute delay lag.
- Disabling the auto Tx mute.
- Controlling the Tx mute manually.

18.2.1 Setting the Auto Tx Mute Delay Lead

Tx Mute Delay Lead is used in Auto Tx Mute to advance the transmitter mute for the specified duration before the last sample is clocked out of the Tx payload buffer.

To set the Tx Mute Delay Lead, type:

```
./phy_ctrl set tx_mute_delay_lead value ↵
```

Where

Parameter	Description
value	Tx mute delay lead value (Normal setting is 60)

To show the Tx Mute Delay Lead, type:

```
./phy_ctrl get tx_mute_delay_lead ↵
```

18.2.2 Setting the Auto Tx Mute Delay Lag

Tx Mute Delay Lag is used in Auto Tx Mute to delay the transmitter mute for the specified duration after the last sample is clocked out of the Tx payload buffer.

To set the Tx Mute Delay Lag, type:

```
./phy_ctrl set tx_mute_delay_lag value ↵
```

Where

Parameter	Description
value	Tx mute delay lag value (Normal setting is 4)

To show the Tx Mute Delay Lag, type:

```
./phy_ctrl get tx_mute_delay_lag ↵
```

18.2.3 Disabling the Auto Tx Mute

The Auto Tx Mute function must always be enabled during normal operation.

To set the Auto Tx Mute, type:

```
./phy_ctrl set auto_tx_mute_disable value ↵
```

Where

Parameter	Description
value	0: auto Tx mute is enabled 1: auto Tx mute is disabled

To show whether the Auto Tx Mute is enabled or not, type:

```
./phy_ctrl get auto_tx_mute_disable ↵
```

18.3 Controlling the Tx Mute Manually

The force_tx_mute parameter allows the mute to be controlled manually. Before using this function, disable the Auto Tx Mute function.

To set the Auto Tx Mute, type:

```
./phy_ctrl set force_tx_mute value ↵
```

Where

Parameter	Description
value	0: Tx mute is off 1: Tx mute is On

To read the Auto Tx Mute, type:

```
./phy_ctrl get force_tx_mute ↵
```

This page has been left intentionally blank.

Section: 19 Frame Configuration

The amount of bandwidth available for user connections is dependent on the frame configuration. This section describes the parameters for configuring the frame and the commands for setting these parameters. The parameters are listed below:

- Frame Code
- Cyclic Prefix (CP) Size
- Forward Error Correction (FEC)
- Duplex Mode
- Transmit/Receive Transition Gap (TTG)
- Receive/Transmit Transition Gap (RTG)
- Contention Slots
- FCH (Frame Control Header) Rate ID

Note:	The Enter key is noted by the following symbol: ↵

For information on each of these parameters, see the following subsections.

19.1 Frame Code

Note:	The frame code cannot be modified once the scheduler is started; i.e. once the “./sched_ctrl set start” command has been executed (set to 0). Therefore if you wish to modify this parameter, you need to reboot the development kit first, change the parameter setting in the base station script and then apply the script. Ensure that the “./sched_ctrl set start” is the last command in the script.

This command can be performed at the base station development kit only. The frame sizes are indicated by frame duration codes as show below.

Table 12 - Number of Symbols per Frame

Frame Duration Codes	Frame Sizes
0	2.5 ms
1	4 ms
2	5 ms
3	8 ms
4	10 ms
5	12.5 ms
6	20 ms

To set the Frame Code, type:

```
./sched_ctrl set frame_code value ↵
```

Where

Parameter	Description
value	(0-6)

To read the frame code, type:

```
./sched_ctrl get frame_code ↵
```

19.2 Cyclic Prefix Size

19.2.1 Setting the Cyclic Prefix (CP) Size at the Base Station

Note:	The cyclic prefix size cannot be modified once the scheduler is started; i.e. once the "./sched_ctrl set start" command has been executed (set to 0). Therefore if you wish to modify this parameter, you need to reboot the development kit first, change the parameter setting in the base station script and then apply the script. Ensure that the "./sched_ctrl set start" is the last command in the script.

To set the cyclic prefix size at the base station, type the following:

```
./sched_ctrl set cp_size value ↵
```

Where

Parameter	Description
value	1/4, 1/8, 1/16, 1/32

To read the cyclic prefix size, type:

```
./sched_ctrl get cp_size ↵
```

Note:	To obtain the CP size at the Subscriber Station in complex samples, multiply the above fractional value by 256.

19.2.2 Setting the Cyclic Prefix Size at the Subscriber Station

To set the cyclic prefix size at the subscriber station, type the following:

```
./phy_ctrl set cp_size value ↵
```

Where

Parameter	Description
value	CP size in complex samples (64, 32, 16, 8)

To read the cyclic prefix size at the subscriber station, type the following:

```
./phy_ctrl get cp_size ↵
```

19.3 FEC (Forward Error Control)

Note:	The FEC cannot be modified once the scheduler is started; i.e. once the "./sched_ctrl set start" command has been executed (set to 0). Therefore if you wish to modify this parameter, you need to reboot the development kit first, change the parameter setting in the base station script and then apply the script. Ensure that the "./sched_ctrl set start" is the last command in the script.

This command can be performed at the base station development kit only.

To set the FEC, type:

```
./sched_ctrl set FEC value ↵
```

Where

Parameter	Description
value	0: Disable FEC 1: Enable FEC

To read the FEC, type:

```
./sched_ctrl get FEC ↵
```

19.4 Duplex Mode

Note:	The duplex mode cannot be modified once the scheduler is started; i.e. once the "./sched_ctrl set start" command has been executed (set to 0). Therefore if you wish to modify this parameter, you need to reboot the development kit first, change the parameter setting in the base station script and then apply the script. Ensure that the "./sched_ctrl set start" is the last command in the script..

This command can be performed at the base station development kit only.

To set the Duplex Mode, type:

```
./sched_ctrl set duplex_mode mode ↵
```

Where

Parameter	Description
mode	fdd tdd

To read the Duplex Mode, type:

```
./sched_ctrl get duplex_mode ↵
```

19.5 TTG (Transmit/Receive Transition Gap)

Note:	The TTG cannot be modified once the scheduler is started; i.e. once the "./sched_ctrl set start" command has been executed (set to 0). Therefore if you wish to modify this parameter, you need to reboot the development kit first, change the parameter setting in the base station script and then apply the script. Ensure that the "./sched_ctrl set start" is the last command in the script.

This parameter can be set at the base station development kit only and is required in TDD (Time Division Duplex) mode only.

To set the TTG, type:

```
./sched_ctrl set ttg value ↵
```

Where

Parameter	Description
value	0… 255 in complex samples. Recommended value: 20.

To read the TTG, type:

```
./sched_ctrl get ttg ↵
```

19.6 RTG (Receive/Transmit Transition Gap)

Note:	The RTG cannot be modified once the scheduler is started; i.e. once the "./sched_ctrl set start" command has been executed (set to 0). Therefore if you wish to modify this parameter, you need to reboot the development kit first, change the parameter setting in the base station script and then apply the script. Ensure that the "./sched_ctrl set start" is the last command in the script.

This parameter can be set at the base station development kit only and is required in TDD mode only.

To set the RTG, type:

```
./sched_ctrl set rtg value↵
```

Where

Parameter	Description
value	0… 255 in complex samples Recommended value: 20.

To show the RTG, type:

```
./sched_ctrl get rtg ↵
```

19.7 Contention Slots

This command can be performed at the base station development kit only and is used to add a contention slot in the uplink sub-frame.

To set the Contention slots, type:

```
./sched_ctrl set contention_slot value↵
```

Where

Parameter	Description
value	0: Disable scheduling of the contention slot 1: Enable scheduling of the contention slot

To read the Contention slots, type:

```
./sched_ctrl get contention_slot ↵
```

19.8 FCH (Frame Control Header) Rate ID

This command will set the Wavesat Rate ID to be used when decoding FCH bursts. BPSK ½ IEEE Rate ID for FCH, Wavesat ID = 6

To set the FCH Rate ID, type:

```
./phy_ctrl set fch_rate_id value↵
```

Where

Parameter	Description
value	6 BPSK ½ IEEE Rate ID for FCH, Wavesat ID = 6

To return the FCH Rate ID, type:

```
./phy_ctrl get fch_rate_id .↵
```

Section: 20 Data Transmission

This section provides the necessary information for users who wish to perform the following:

- Start the scheduler at the base station.
- To pause the scheduler.

Note:	The Enter key is noted by the following symbol: ↵

20.1 Start

The following command must be performed at the base station to start the scheduler. The scheduler is part of the MAC layer and is used to allocate bandwidth for user connections. The value entered in this command is used to set the transmission delay between frames, which is the product of this value and the operating system (O/S) clock setting.

For example, if the O/S clock is 10 ms, then setting this value to 2 would result in a transmission delay of 20ms.

To set the start delay, type:

```
./sched_ctrl set start value ↵
```

Where

Parameter	Description
value	0: To set the value of the transmission delay between frames.

Note:	This value is for factory tests. **Users must leave this value set to 0.** Other values will reduce the development kit speed.

20.2 Pause

Once the command "./sched_ctrl set start" is executed, to pause the scheduler and thus interrupt the data transmission between the base station and the subscriber station, you can use the following command at the base station.

To pause the scheduler, type:

```
./sched_ctrl set pause value ↵
```

Where

Parameter	Description
value	0: Restarts the transmission that was previously paused 1: Pauses the transmission

To read the pause, type:

```
./sched_ctrl get pause ↵
```

This page has been left intentionally blank.

Section: 21 Interrupts

DM256 employs the following interrupts:

- Tx data buffer ready.
- Rx data buffer ready.
- End of Tx Frame.
- End of Rx Frame.
- Internal Header Error.
- Frame Configuration Error.
- Tx Payload Buffer Reset.
- FCH (Frame Control Header) Symbol Received.
- Rx buffer overflow.
- Interrupt statistic counters.
- Tx FIFO Underrun Interrupt

Note:	When the interrupt Control/Status Register is set, it's not possible to read the configuration o the interrupt. When the register is read, all interrupt values are set to 0.

For the description of each of the interrupts and their corresponding commands, see the following subsections.

Note:	The Enter key is noted by the following symbol: ↵

21.1 Tx Data Buffer Ready

To set the Tx Data Buffer Interrupt, type:

```
./phy_ctrl set tx_data_ready value ↵
```

Where

Parameter	Description
value	0= Enabled 1= Disabled

21.2 Rx Data Buffer Ready

To set the Rx Data Buffer Ready Interrupt, type:

```
./phy_ctrl set rx_data_ready value ↵
```

Where

Parameter	Description
value	0= Enabled 1= Disabled

21.3 End of Tx Frame

To set the End of Tx Frame Interrupt Header Error, type:

```
./phy_ctrl set tx_frame value ↵
```

Where

Parameter	Description
value	0= Enabled 1= Disabled

21.4 End of Rx Frame

To set the End of Rx Frame interrupt, type:

```
./phy_ctrl set rx_frame value ↵
```

Where

Parameter	Description
value	0= Enabled 1= Disabled

21.5 Internal Header Error

To set the Internal Header Error interrupt, type:

```
./phy_ctrl set int_header_err value ↵
```

Where

Parameter	Description
value	0= Enabled 1= Disabled

21.6 Frame Configuration Error

To set the Frame Configuration Error interrupt, type:

```
./phy_ctrl set cfg_symb_err value ↵
```

Where

Parameter	Description
value	0= Enabled 1= Disabled

21.7 Tx Payload Buffer Reset

To set the Rx Payload Buffer Reset Interrupt, type:

```
./phy_ctrl set tx_fifo_reset value ↵
```

Where

Parameter	Description
value	0= Enabled 1= Disabled

21.8 FCH Symbol Received

To set the FCH Symbol Received Interrupt, type:

```
./phy_ctrl set config_symb value ↵
```

Where

Parameter	Description
value	0= Enabled 1= Disabled

21.9 Rx Buffer Overflow

To set the Rx Buffer Overflow, type:

```
./phy_ctrl set rx_buff_overflow value ↵
```

Where

Parameter	Description
value	0= Enabled 1= Disabled

21.10 Reset IRQ (Interrupt statistics) Counters

To set the IRQ, type:

```
./phy_ctrl set reset_irq_stats value ↵
```

Where

Parameter	Description
`value`	0= No reset is performed. 1= All interrupts are set to 0.

21.11 Tx FIFO Underrun Interrupt

This command enables generation of an interrupt when the DM 256 is scheduled to generate an OFDM symbol for Tx and there is insufficient data in the Tx FIFO to perform the operation. The symbol will be padded with 0xFF.

To set the Tx FIFO Underrun interrupt, type:

```
./phy_ctrl set tx_fifo_underrun value ↵
```

Where

Parameter	Description
`value`	0= Disable Tx FIFO underrun. 1= Enable Tx FIFO underrun.

Section: 22 Direct Memory Access (DMA)

Direct Memory Access (DMA) is used to transfer the payload buffer between DM 256 and the CPU. The DMA Engine on the CPU must be enabled when transferring data in DMA mode.. See the following subsections for the DMA request commands or DMA commands for:

- Tx Payload.
- Rx Payload.
- Tx Frame Configuration.
- Rx Frame Configuration.
- Rx Payload DMA transfer width.
- Tx Payload Buffer Content Size.
- Rx Payload Buffer Content Size

Note:	The Enter key is noted by the following symbol: ↵

22.1 Tx Payload Request

To set DMA request for Tx payload, type:

```
./phy_ctrl set tx_dma value ↵
```

Where

Parameter	Description
value	0= Disabled 1= Enabled

To read the DMA request status for Tx payload, type:

```
./phy_ctrl get tx_dma ↵
```

22.2 Rx Payload Request

To set DMA request for Rx payload, type:

```
./phy_ctrl set rx_dma value ↵
```

Where

Parameter	Description
value	0= Disabled 1= Enabled

To read the DMA request status for Rx payload, type:

```
./phy_ctrl get rx_dma ↵
```

22.3 Rx Payload DMA Transfer Width

This procedure is used to define the size of the transfer bus. It is by default set to byte since this allows all multiples to be allowed in the various burst profiles. However if a customer is sure that the information will be in multiples of 2 or 4, the value could be changed to the value of the register to half word (16 bits) or word (32 bits).

To set the Rx payload DMA transfer width, type:

```
./phy_ctrl set rx_dma_transf_width value ↵
```

Where

Parameter	Description
[value]	0 = Word 1 = Byte (default) 10 = Half word 11 = Word

Note:	The Rx payload DMA transfer width is set by default to byte since it is possible to transfer an even number of bytes in a frame.

To verify the Rx payload DMA transfer width, type:

```
./phy_ctrl get rx_dma_transf_width ↵
```

22.4 Tx Payload Buffer Content Size

To retrieve the content size of the Tx payload buffer, type:

```
./phy_ctrl get tx_fifo_count ↵
```

22.5 Rx Payload Buffer Content Size

To retrieve the content size of the Rx payload buffer, type:

```
./phy_ctrl get rx_fifo_count ↵
```

This page has been left intentionally blank.

Section: 23 List of Terms and Definitions

ADC	Analog to Digital Converter
AFC	Automatic Frequency Correction
AGC	Automatic Gain Control
ALC	Automatic Level Control
ALU	Arithmetic Logic Unit
ARQ	Automatic Repeat Request
BR	Bandwidth Request
Baseband	Baseband is the original band of frequencies produced by a transducer or any other signal initiating device, prior to initial modulation. Bandwidth is centered at DC (IF freq = 0 Hz)
Base Station (BS)	The equipment that provides connectivity, management, and control of the Subscriber Station (SS).
BER	Bit Error Rate
BPSK	Binary Phase Shift Keying
Broadband	Having instantaneous bandwidths greater than around 1 MHz and supporting data rates greater than about 1.5 Mb/s.
Broadband Wireless Access (BWA)	Wireless access in which the connection(s) capabilities are broadband.
Burst Profile	Set of parameters that describe the uplink or downlink transmission properties associated with an interval usage code. The parameters include symbol type, modulation rate and channel coding.
BW	Bandwidth
Center Frequency	The center of the frequency band in which the transmission occurs. Center frequency of the received or transmitted Bandwidth.
CFG Word	It is the same thing as DL/UL CFG & SCH Frame but is a lot easier to say. Note that "map" may be used synonymously because they are simply different forms of the same thing.
Cheetah	Hardware Medium Access Control layer
CID	Connection Identifier
CINR	Carrier to Interference and Noise Ratio
CMG	Channel Measurement Gap
Connection Identifier CID	A unidirectional MAC address that identifies a connection to equivalent peers in the MAC of the Base Station (BS) and Subscriber Station (SS).
COFDM	Coded Orthogonal Frequency Division Multiplexing
Cyclic Prefix (CP) Size / CP Time	Guard samples per symbol. Possible values are 8, 16, 32 and 64 based on an FFT size of 256 samples per symbol.
CP	Cyclic Prefix
CPE	Customer Premise Equipment as known as Subscriber Station
CRC	Cyclic Redundancy Check
CS	Convergence Sub-layer
DAC	Digital to Analog Converter
DCD	Downlink Channel Descriptor
DFS	Dynamic Frequency Selection
DIUC	Downlink Interval Usage Code
DL	Downlink
DLFP	Downlink Frame Prefix
Downlink (DL)	The direction from the Base Station (BS) to the Subscriber Station (SS).

Downlink Channel Descriptor (DCD)	A MAC message that describes the physical layer characteristics of a downlink channel and is transmitted by the Base Station at a periodic interval.
Downlink Interval Usage Code (DIUC)	An interval usage code used in the downlink. See Interval Usage Code (IUC).
Downlink Map (DL-MAP)	A MAC management message generated at the Base Station and broadcast to all Subscriber Stations. It is software dependent. It describes the burst profiles (configuration and scheduling) of the downlink sub-frame. The DL-MAP is used to create the content of the Rx Frame Configuration Buffer, which is required by each Subscriber Station in order to deconstruct the downlink sub-frame received from the Base Station.
DL-MAP IE	Downlink MAP Information Element
DL/UL CFG & SCH Frame (Descriptor)	It is hardware dependent. It describes the burst profiles (configuration and scheduling) of the downlink sub-frame.
DRAM	Dynamic Random Access Memory
EEPROM	Electronically Erasable Programmable Read-Only Memory
Fast-Fourier Transform (FFT) size	Referred as the "useful" symbol time. The FFT size is the smallest power of two greater than the number of sub-carriers used, which is 200. Therefore, the FFT size is 256 points per symbol.
FCH	Frame Control Header
FDD	Frequency Division Duplex or Duplexing
FEC	Forward Error Correction
FFT	Fast Fourier Transform
FIFO	First In First Out
Frame	A structured set of data that is transmitted over the air interface for a specific duration. DM 256 can be programmed to use the following frame sizes: 2.5, 4, 5, 8, 10, 10, 12.5 and 20 ms. This word can be used synonymously with "Message".
Frame Descriptor	Information stored in a buffer and that describes the burst profiles (configuration and scheduling) of a frame.
Frequency Division Duplex (FDD)	A duplex scheme in which uplink and downlink transmissions use different frequencies but occur typically at the same time.
FSH	Fragmentation Sub-header
Guard Interval (G)	Ratio of CP Size (a.k.a. CP Time) to FFT size. Possible values are 1/32, 1/16, 1/8 and 1/4.
Half-Duplex Frequency Division Duplex (H-FDD)	A duplex scheme in which uplink and downlink transmissions use not only different frequencies but also occur at different times.
HCS	Header Check Sequence
HEC	Header Error Check
HFDD	Half-Duplex Frequency Duplexing
HSAR	Hardware Segmentation And Re-assembly
HT	Header Type
LOS	Line of Sight
I	In phase
IE	Information Element
IEEE	Institute of Electrical and Electronics Engineers
Interval Usage Code (IUC)	A code identifying a particular burst profile that can be used by a downlink or uplink transmission interval.
IP	Internet Protocol
I/Q	Inphase/Quadrature
IUC	Interval Usage Code
LSB	Least Significant Bit
MAC	Medium Access Control Layer

Message	Describe a group of data, which is a complete entity of varying length, but may be transported over one or more packets. For example: a MAC message may be transported over one or more packets, which may be transmitted over one or more OFDM symbols.
MSB	Most Significant Bit
NFS	Network File System
NLOS	Non Line of Sight
OFDM	Orthogonal Frequency Division Multiplexing
Packet	This word is used to describe a fixed size group of data. This typically refers to a transport packet. 802.16 - 2004 defines many packet sizes depending on modulation and coding rate.
PAPR	Peak to Average Power Ratio
Physical Layer (PHY)	The lowest layer within the OSI Network Model. It deals primarily with transmission of the raw bit stream over the Physical transport medium. In the case of wireless transmission, the transport medium is free space. This layer defines parameters such as data rates, modulation method, signaling parameters, transmitter/receiver synchronization etc. Within an actual radio implementation, the PHY corresponds to the radio front end and baseband signal processing sections.
PMP	Point-to-MultiPoint
PQFP	Plastic Quad Flat Pack
Q	Quadrature
QAM	Quadrature Amplitude Modulation
QoS	Quality of Service
QPSK	Quadrature Phase-Shift Keying
Protocol Data Unit (PDU)	This is a group of data strictly defined by the protocol. It is a formatted group of data exchanged between two adjacent layers of the OSI protocol. In the downward direction, it is the data to be transmitted to the lower layer. On the upward direction, it is the data received from the lower layer. For example, the PHY PDU is the formatted burst of data transmitted over the air.
Receive/Transmit Transition Gap (RTG)	A time interval between the uplink sub-frame and the following downlink sub-frame in time division duplexing (TDD). This time interval allows the Base Station (BS) to switch from receive to transmit mode and Subscriber Stations (SS) to switch from transmit to receive mode. During this time interval, the BS and SS are not transmitting modulated data but simply allowing the BS transmitter carrier to ramp up, the transmit/receive (Tx/Rx) antenna switch to actuate, and the SS receiver section to activate. RTG is not applicable to frequency division duplexing (FDD).
REQ	Request
RF Center Frequency	The center of the frequency band in which a base station (BS) or SS is intended to transmit.
RID	Rate ID
RNG-RSP	Ranging -Response
RS	Reed–Solomon
RTG	Receive/Transmit Transition Gap
Rx	Receiver
Sampling Frequency	The Sampling Frequency is calculated as the product of the nominal channel bandwidth and a sampling factor of 8/7.
SAR	Segmentation And Re-assembly
SDRAM	Synchronous Dynamic Random Access Memory
SERDES	Serialize/De-Serialize
Service Data Unit (SDU)	A formatted group of data exchanged between two adjacent layers of the OSI protocol. In the downward direction, it is the data received from the higher layer. In the upward direction, it is the data to be transmitted to the higher layer.
SMAC	Software Medium Access Control Layer
SNR	Signal to Noise Ratio
SPI	Serial Peripheral Interface

SSAR	Software Segmentation And Re-assembly
Subscriber Station (SS)	A generalized equipment set providing connectivity between subscriber equipment and a Base Station (BS).
SUI Channel Models	Stanford University Interim Channel Models
Symbol	Describe a coded, modulated group of data, usually for transmission or reception. This typically refers to an OFDM symbol. For example: REF symbol, FCH symbol, DATA symbol, etc.
Symbol Length	Number of samples in a symbol. It is the sum of FFT size and CP Size. Possible values are 264, 272, 288 and 320.
Symbol Time	The Symbol Time is the number of samples in a symbol (Symbol Length) divided by the Sampling Frequency.
Time Division Duplex or Duplexing (TDD)	A duplex scheme where uplink and downlink transmissions occur at different times but may share the same frequency.
Time Division Multiplexing (TDM) Burst	A contiguous portion of a TDM data stream using PHY parameters, determined by the Downlink Interval Usage Code (DIUC), that remain constant for the duration of the burst. TDM bursts are not separated by gaps or preambles.
Time Division Multiple Access (TDMA) Burst	A contiguous portion of the uplink or downlink using PHY parameters, determined by the Downlink Interval Usage Code (DIUC) or Uplink Interval Usage Code (UIUC), that remain constant for the duration of the burst. TDMA bursts are separated by preambles and are separated by gaps in transmission if subsequent bursts are from different transmitters.
Type/Length/Value (TLV)	A formatting scheme that adds a tag to each transmitted parameter containing the parameter type (and implicitly its encoding rules) and the length of the encoded parameter.
TO	Transmission Opportunity
Transmit/Receive Transition Gap (TTG)	A time interval between the downlink sub-frame and the following uplink sub-frame in time division duplexing (TDD). This time interval allows the Base Station (BS) to switch from transmit to receive mode and subscriber stations (SS) to switch from receive to transmit mode. During this time interval, the BS and SS are not transmitting modulated data but simply allowing the BS transmitter carrier to ramp down, the transmit/receive (Tx/Rx) antenna switch to actuate, and the BS receiver section to activate. TTG is not applicable to frequency division duplexing (FDD).
Tx	Transmitter
Uplink (UL)	The direction from a Subscriber Station to the Base Station (BS).
Uplink Channel Descriptor (UCD)	A MAC management message that describes the physical layer characteristics of an uplink channel and is transmitted by the Base Station at a periodic interval.
Uplink Interval Usage Code (UIUC)	An interval usage code used in the uplink. See Interval Usage Code (IUC).
Uplink Map (UL-MAP)	A MAC management message generated at the Base Station and broadcast to all Subscriber Stations. It describes the burst profiles (configuration and scheduling) of the uplink sub-frame. The UL-MAP is used to create the content of the Tx Frame Configuration Buffer which is required by each Subscriber Station in order to construct the uplink sub-frame to be transmitted to the Base Station. A set of information that defines the entire access for a scheduling interval.
UL-MAP IE	Uplink MAP Information Element

Appendix A: vi Commands

To write or modify scripts for the DM256, you need to use vi (**vi**sual editor), which is a full-screen editor. This section describes only the commands that you might commonly use.

Note:	vi commands are **case-sensitive.**

A.1 Starting vi

vi [filename]	To create or edit a file, type vi followed by the filename. If the file exists, it will be displayed. If the file does not exist, then the empty screen of a new file is displayed.

A.2 Exiting vi

:q <Return>	Quit vi after saving any changes made
:q! <Return>	Quit vi without saving any changes

A.3 Moving the Cursor

j or <Return>	Move cursor down by one line
k	Move cursor up by one line
h or <Backspace>	Move cursor left by one character
l or <Space>	Move cursor right by one character
0 (zero)	Move cursor to the start of the current line
$	Move cursor to the end of the current line

Note:	The arrow keys can also be used to move the cursor.

A.4 Manipulating the Screen

^f	Move down (forward) by one screen
^b	Move up (backward) by one screen
^d	Move down one half screen
^u	Move up one half screen
^r	Refresh the screen and remove deleted lines

Note:	The symbol ^ before a letter indicates that the <Ctrl> key must be held down while the letter key is pressed.

A.5 Adding Text

i	Insert text before the cursor position
I	Insert text at beginning of the current line
a	Append text after the cursor position
A	Append text to end of the current line
o	Open and put text in a new line below the current line
O	Open and put text in a new line above the current line

Note:	Pressing each of the above commands places the vi editor in Insert mode. In this mode, every character you type is added to the displayed file. To exit from the insert mode; press the <Esc> key.

A.6 Deleting Text

x	Delete the single character in the cursor position
dd	Delete the entire current line

A.7 Replacing Text

r	Replace a single character. After pressing this command key, the next letter you type will replace the letter in the current cursor position.
R	Replace characters starting from the current cursor position. After pressing this command key, the next letter you type will replace the letter in the cursor position until you press the <Esc> key.

A.8 Copying and Pasting Text

yy	Copy the current line into the buffer
p	Paste the line in the buffer after the current line

A.9 Undoing the Last Action

u	Undo your last action. This command acts like a toggle by undoing and redoing your most recent action. You cannot go back more than one step.

A.10 Searching Text

/string	Search forward for the occurrence of “string” in the text
?string	Search backward for the occurrence of “string” in the text
n	Search forward for the next occurrence of “string” in the text
N	Search backward for the next occurrence of “string” in the text

A.11 Determining the Number of Lines

:=	Returns, at the bottom of the screen, the total number of lines in the displayed file

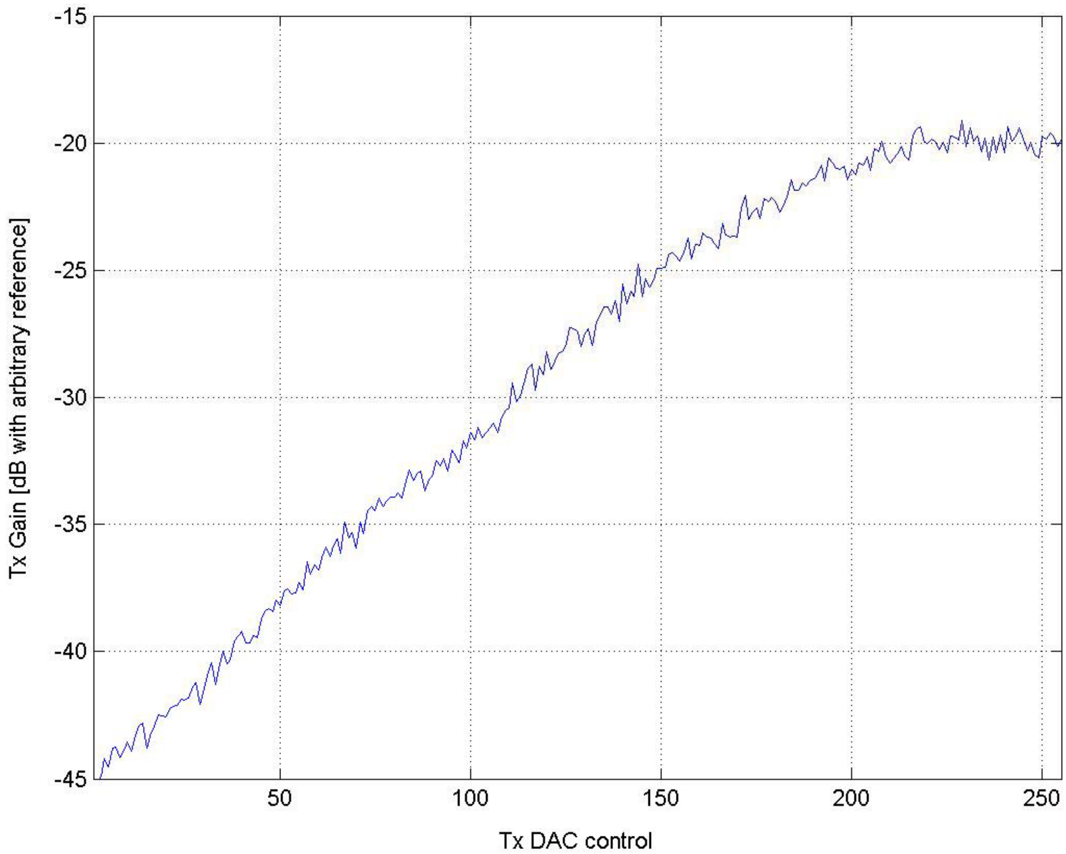
A.12 Creating a New File Based on the Displayed File

:w [new file] <Return>	Write the content of the displayed file to a new file named "new file"

This page has been left intentionally blank.

Appendix B: Calibration and Test Results

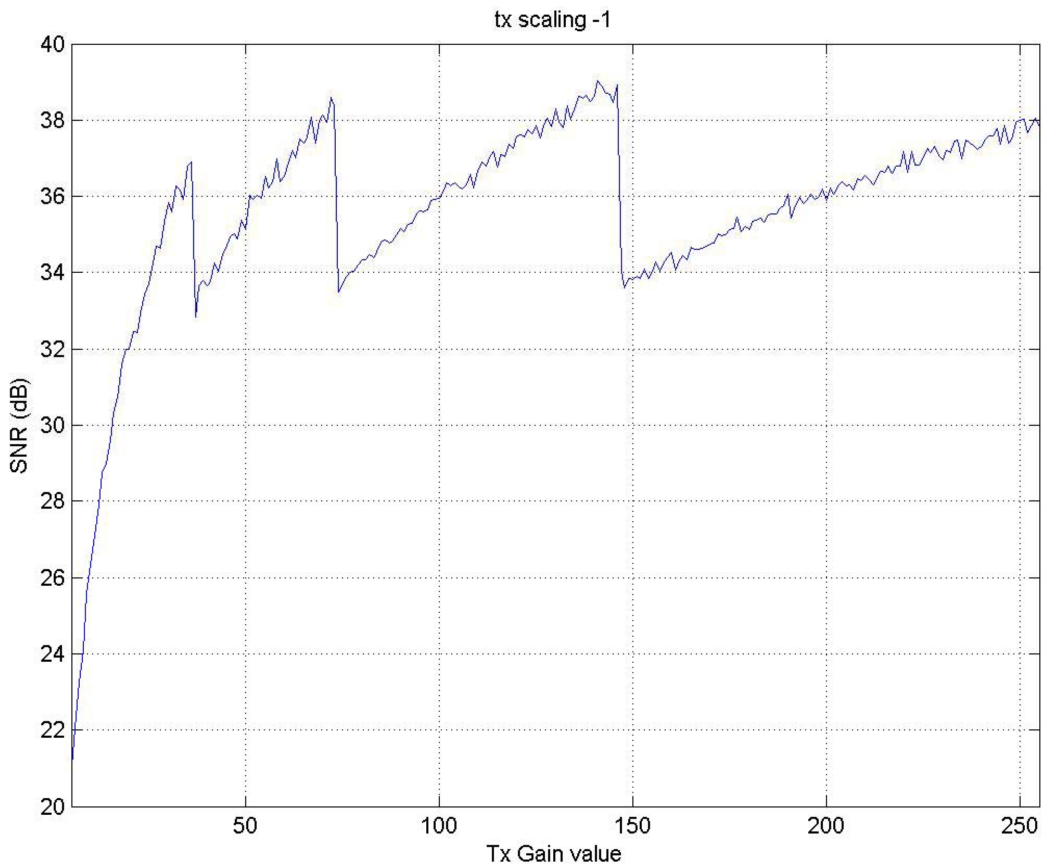
B.1 Tx Gain versus Tx DAC



000012

Figure 25 - Tx Gain versus Tx DAC Graph

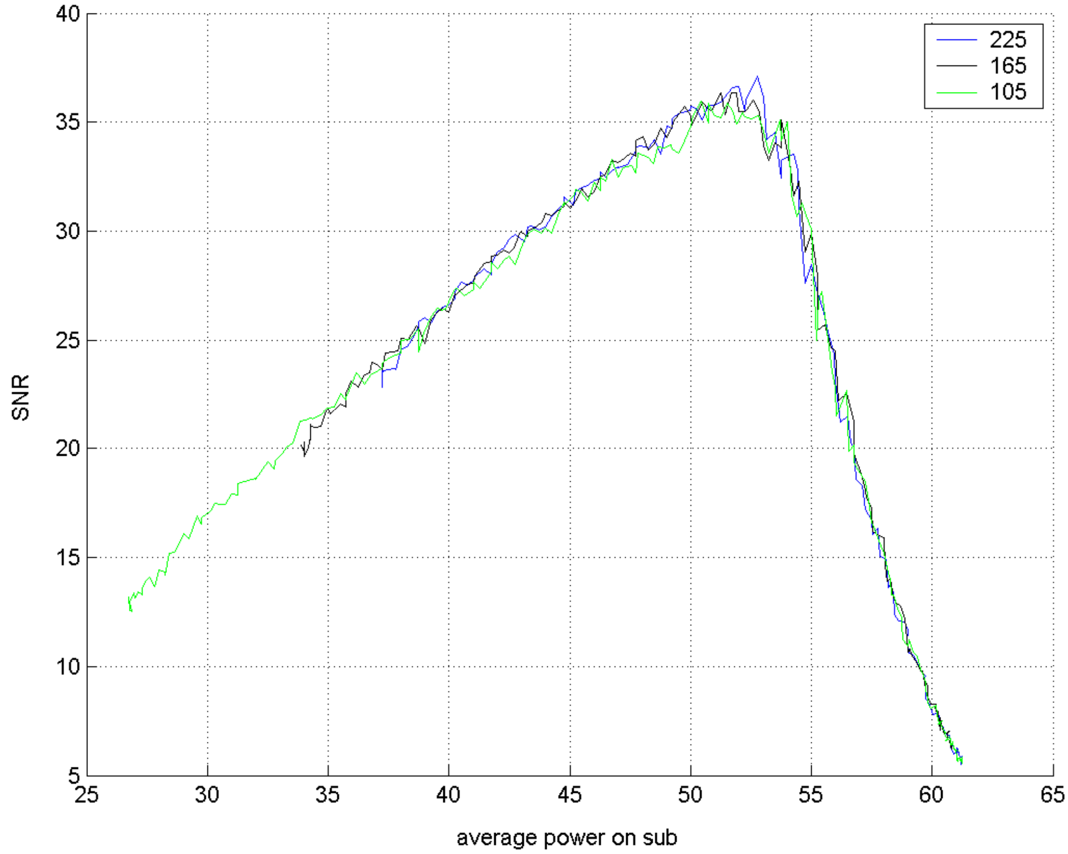
B.2 SNR versus Tx Digital Gain



000013

Figure 26 - SNR versus Tx Digital Gain Graph

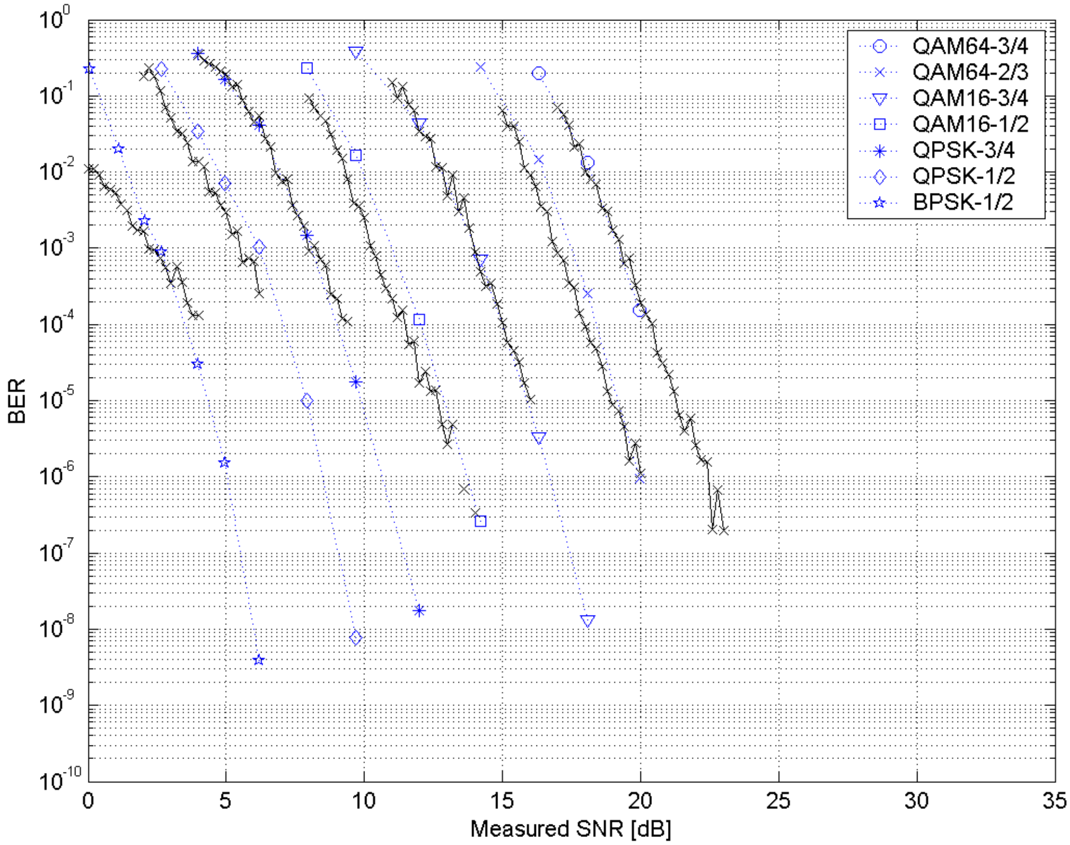
B.3 SNR versus Rx Power



000014

Figure 27 - SNR versus Rx Power Graph

B.4 BER versus SNR



000015

Figure 28 - BER versus SNR Graph

Note:	In the above graph, data which is in black (more tightly plotted together, denoted by Xs) is simulated data while actual measured data is in blue (more spaced out, denoted in stars, diamonds, squares, triangles, circles,…).

Appendix C: Source Code Description

Table 13 - Directory Structure Description

anacondaDrv	Contains the device driver of the anaconda analog board.
bin	Resulting directory target of the makefile. After compilation, it contains all the kernel modules and the user space applications.
ucbin	Resulting directory target of the makefile compiled against the micro libC.
macLink	Contains the link control layer code for MAC Messaging Processing, Initialization, Ranging and Network Entry
modMain	Contains the kernel module entry point of the network driver.
phyDriver	Contains the OFDM PHY driver interface code. It also contain the code responsible for programming the FPGA version of the PHY, the interrupt handler and the PHY DMA interface.
sched	Contains the source code responsible for user bandwidth allocation and PHY config word programming.
services	Contains generic functions.
macDb	Contains the source code responsible for storing connection configuration and allocation.
ofdmNetDrv	Contains the source code used to transmit and receive data through the OFDM PHY.
scripts	Contains the scripts used to run the base station and subscriber station development kit.
apps:	
anaconda_ctrl:	User app used to control the anaconda analog board.
fpga_prog:	User app used to program the FPGA with a binary file.
macdb_ctrl:	User app used to add, modify and remove connections on the base station.
ofdm_stats:	User app used to fetch the development kit statistics.
phy_ctrl:	User app used to interface with the PHY registers.
sched_ctrl:	User app used to interface with the scheduler.
ss_linktest_ctrl:	User app used to add and remove connections on the subscriber station.
tools:	Contains libraries used to build the source. It also contains tools used to build the flash version of the development kit (like uClibc and busybox).

C.1 Compiling the Sources

C.1.1 Kernel source

The following steps are required to build the linux kernel.

1. Go to the root of the kernel source.
2. Make menuconfig
3. Without changing anything, exit and save the configuration.
4. Build the kernel: make clean dep pImage.

The resulting kernel image resides in arch/ppc/boot/ppc_load/. Since the Dragon development kit use PPCBoot for it's boot functions, the target pImage must be used as a kernel target.

C.1.2 Device Driver

The following steps are required to compile the dragon development kit source code:

1. Go into dm216 directory.
2. Create a softlink to the kernel source: ln -s <kernel_src_path> kernel_src
3. Build the code: make dep all

A soft link called kernel_src needs to be created in the root of the dm216 directory. The link must point to the dragon development kit kernel source. The resulting build including the drivers and the applications resides in the bin directory.

The makefile used to compile the entire source tree supports the following makefile targets:

- all: Build everything.
- depend: build the dependencies for all the modules and applications.
- clean: Clean all the objects (.o).

C.1.3 Cross-compiling

To build the driver in a cross platform environment, the variable CROSS in the file Rules.make must be set to match your cross-compiler's prefix.

The current OFDM development kit was developed using GNU gcc version 3.2. The following parameters were used to build the cross compiler:

> *Configured with: ../package/configure --build=i686-linux --host=i686-linux –target=powerpc-linux --prefix=/opt/timesys/linux/4.0/toolchains/ppc4xx-linux --enable-threads=posix –program-transform-name='s,^,ppc4xx-linux-,' --enable-languages=c,c++ --enable-shared --with-gnu-as --with-gnu-ld --nfp --with-stabs*
> *Thread model: posix*
> *gcc version 3.2*

The compiler GNU gcc can be obtained at http://gcc.gnu.org/. GNU gcc also requires the latest binutils package. It can be obtained at ftp://ftp.kernel.org/pub/linux/devel/binutils.

C.1.4 LibC: Linkage

Since the size of the complete libc is around 24 MB, a special version of libc is required to run the dragon applications from the flash. The micro libc (uClibc) library is used for this purpose. To compile using the micro libc, the following makefile parameter is required: make LIB_C=uClibc. uclibc can be obtained at http://www.uclibc.org/.

C.2 Device drive

C.2.1 Char device names

The compiled device drivers require the following char device names to be added to the /dev directory of the root file system:

mknod /mnt/jffs/dev/anaconda c 251 0

mknod /mnt/jffs/dev/i2c-0 c 89 0

mknod /mnt/jffs/dev/ofdm_net_stats c 253 0

mknod /mnt/jffs/dev/ofdm_mac_sched c 253 1

mknod /mnt/jffs/dev/macdb c 253 2

mknod /mnt/jffs/dev/ofdm_phy_ctrl c 254 0

mknod /mnt/jffs/dev/ofdm_phy_prog c 254 1

mknod /mnt/jffs/dev/ofdm_phy_loopback c 254 2

This page has been left intentionally blank.

Appendix D: Network File System (NFS) mode

The Network File System (NFS) mode is used when the load resides on an NFS server instead of inside the flash memory of the development kit.

D.1 Powering the Development kit and Logging In

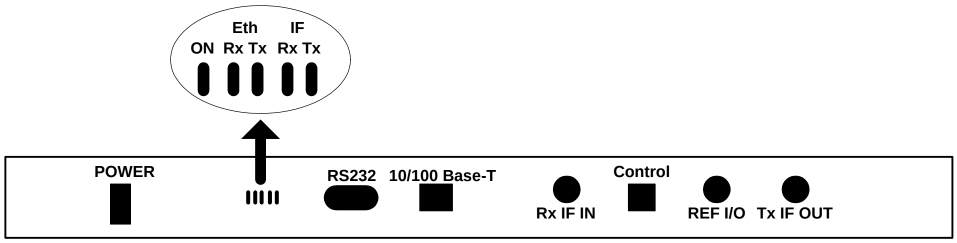


Figure 29 - Console Interface

To power and log into each development kit, proceed as follows:

1. Turn the development kit On by toggling the POWER switch on the faceplate of the development kit.

   The ON LED illuminates.

   When the booting process is completed, the following prompt appears:

   ```
   aa.bb.cc.dd login:
   ```

   Where

Parameter	Description
aa.bb.cc.dd	Is the IP address of the development kit

2. Enter the username by typing:

   ```
   root ↵
   ```

Note:	The Enter key is noted by the following symbol: ↵

3. The following prompt appears:

   ```
   bash-2.05#
   ```

   You are now ready to type the desired commands.

-End-

D.2 Modifying the Bootstrap

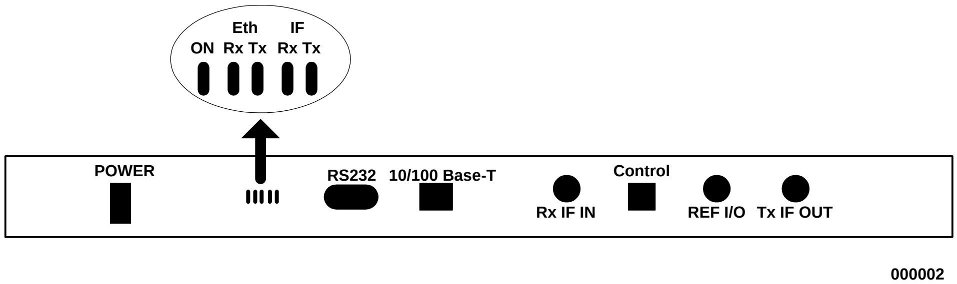


Figure 30 - Console Interface

To modify the bootstrap of each development kit, proceed as follows:

1. Turn the development kit On by toggling the POWER switch on the faceplate of the development kit.

   The ON LED illuminates.

2. While the booting process is occurring, press any key to stop the process.
3. To display the bootstrap settings, type:

   `Print ↵`

The bootstrap settings such as the following appear:

```
bootdelay=3
baudrate=9600
loads_echo=1
ethaddr=00:00:00:06:00:cc
ofdmaddr=00:00:00:06:01:cc
ipaddr=10.1.10.150
serverip=10.1.10.129
bootcmd=bootm 0x8010000
bootargs=root=/dev/nfs rw
nfsroot=10.1.10.129:/opt/timesys/linux/4.0/ep405/rfs ip=10.1.10.150:
10.1.10.129:255.255.0.0 console=ttyS0,9600
stdin=serial
stdout=serial
stderr=serial
```

4. To change the Ethernet address, type:

   `setenv ethaddr 00:00:00:xx:yy:zz ↵`

   Where

Parameter	Description
xx	is the board revision
yy	00 for Ethernet and 01 for OFDM
zz	hexadecimal value of the serial number of the board

-Continue-

5. To change the OFDM address, type:

```
setenv ofdmaddr 00:00:00:xx:yy:zz ↵
```

Where

Parameter	Description
xx	is the board revision
yy	00 for Ethernet and 01 for OFDM
zz	hexadecimal value of the serial number of the board

6. To change the server IP address, type:

```
setenv serverip aa.bb.cc.dd ↵
```

Where

Parameter	Description
aa.bb.cc.dd	is the IP address of the server used for file transfer

7. To change the development kit IP address, type:

```
setenv ipaddr aa.bb.cc.dd ↵
```

Where

Parameter	Description
aa.bb.cc.dd	Is the IP address of the development kit

8. To change the boot arguments used when booting the development kit, type:

```
setenv bootargs root=/dev/nfs rw
nfsroot=10.1.10.129:/opt/timesys/linux/4.0/ep405/rfs
ip=aa.bb.cc.dd:ee.ff.gg.hh:255.255.0.0 console=ttyS0,9600 ↵
```

Note:	Ensure that you have typed in the entire command until the symbol ↵ where you would press the enter key.

Where

Parameter	Description
aa.bb.cc.dd	is the development kit IP address
ee.ff.gg.hh	is the server IP address
255.255.0.0	is the netmask

-End-

D.3 Upgrading the Kernel

To upgrade the kernel (pImage), proceed as follows:

1. Before you put the kernel on the development kit, unprotect the flash by typing the following command:

   ```
   =>protect off all ↵
   ```

2. Erase the content of the flash by typing the following command:

   ```
   =>erase all ↵
   ```

3. Perform an TFTP operation to download the kernel load (pImage) to flash by typing the following command:

   ```
   =>tftp 0x8010000 pImage ↵
   ```

4. Configure the bootcmd parameter by typing the following command:

   ```
   =>setenv bootcmd bootm 0x8010000 ↵
   ```

   Where

Parameter	Description
bootm	is the command to boot the kernel
0x8010000	is the address where the kernel is stored

5. If you have made any modifications, save them by typing:

   ```
   saveenv ↵
   ```

6. Reset the development kit by restarting it or by typing the following command:

   ```
   reset ↵
   ```

-End-

D.4 Listing Files & Scripts

To list files in the directory located on your server, type:

```
ls ↵
```

The list of required files is given below. i.e. scripts, SW loads (xxx.o) and FPGA load.

- Software modules:
  - SubNetDrv.o (only used if configured in subscriber mode)
  - BaseNetDrv.o (only used if configured in base station mode)
  - PhyDriver.o (PHY driver module)
  - anacondaDrv.o (Analog control module)
- Functions:
  - phy_ctrl (Function related to the PHY)
  - anaconda_ctrl (Function related to the Analog board)
  - fpga_prog (Function to program the FPGA)
  - sched_ctrl (Function used for the scheduling)
  - macdb_ctrl (Function used for the management of the connection)
  - ofdm_stats (Function used for the statistics)
  - ss_linktest_ctrl (Function used for the link management)

D.5 Upgrading the FPGA File

To download the FPGA file, proceed as follows:

1. Locate the directory on the NFS server where the load resides. The command “printenv” will display the path as “nfsroot” parameter.
2. When you have located the directory, copy the new FPGA load either in *.bin or *.gz format.

D.6 Upgrading the Command Interface Software

1. Locate your home directory on the NFS server. The command “printenv” will display the path as “nfsroot” parameter.
2. When you are in your home directory, create a soft link by typing:

```
ln -s dir_name link_name ↵
```

Where

Parameter	Description
dir_name	is located on the server 10.1.10.129 in /opt/timesys/linux/4.0/ep405/rfs/dm216/vx.x
link_name	is usually called “load”.

This page has been left intentionally blank.

Appendix E: RS-232 Cable

Straight RS-232 Cable is required to connect your computer/console to the development kits.

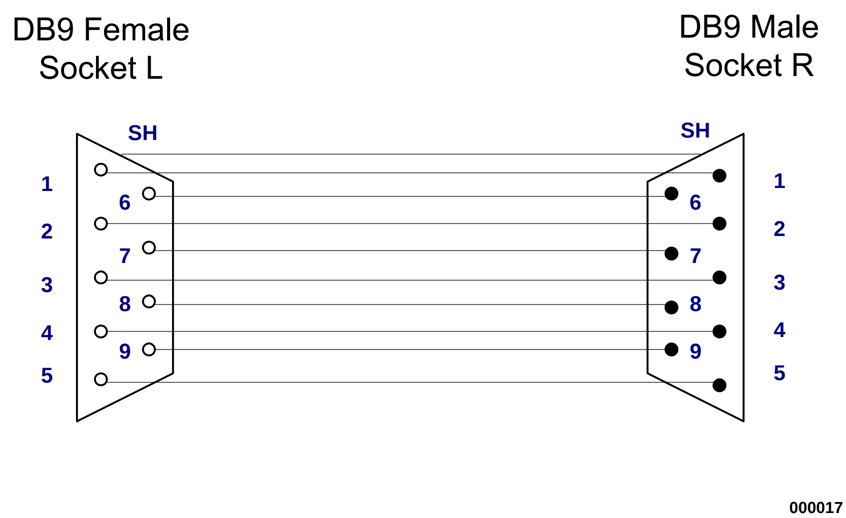


Figure 31 - DB9 Female to DB9 Male Socket Diagram for RS-232 Cable

Table 14 - DB9 Female to DB9 Male Socket Pin to Pin Connection for RS-232 Cable

Female DB9 Socket DTE	Male DB9 Socket DCE
L-1	R-1
L-2	R-2
L-3	R-3
L-4	R-4
L-5	R-5
L-6	R-6
L-7	R-7
L-8	R-8
L-9	R-9
L-SH	R-SH

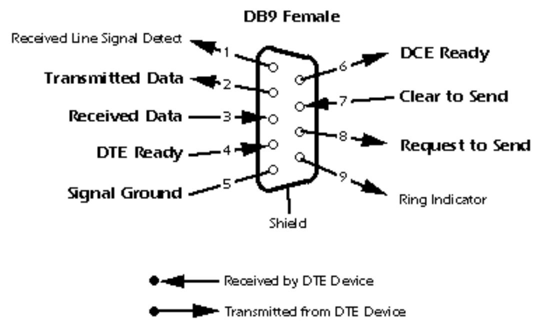


Figure 32 - DB9 Female Socket Pinout for RS-232 Cable

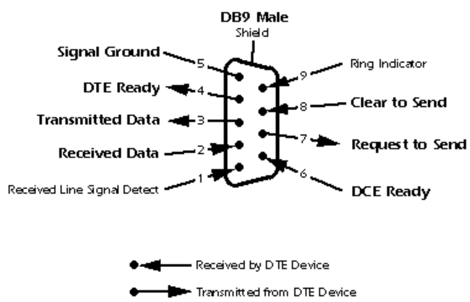


Figure 33 - DB9 Male Socket Pinout for RS-232 Cable

The above figures and table show a 9-pin DTE-to-DCE serial cable that would result if the EIA232 (RS 232) standard were strictly followed. All 9 pins plus shield are directly extended from DB9 Female to DB9 Male. There are no crossovers or self-connects present. Use this cable to connect development kits, printers, or any device that uses a DB9 connector to a PC's serial port.

This cable may also serve as an extension cable to increase the distance between a computer and serial device.

Caution:	Do not exceed 25 feet separation between devices without a signal booster!

Appendix F: Development Kit Booting Scripts

The following are the Base Station and Subscriber Station Booting Scripts for the Development Kits. You can find these scripts on your Development Kits or on the CD provided with your Development Kits.

F.1 Base Station Booting Script

To obtain the booting scripts type the following command at the Base Station, type:

`cat base` ↵

```
# cat base
#/bin/sh
insmod anacondaDrv.o
insmod phyDriver.o
./anaconda_ctrl set sample_clk 40000000
./fpga_prog v2r0b71d6k.bin
./phy_ctrl set tx_mute_delay_lead 60
./phy_ctrl set tx_mute_delay_lag 4
./phy_ctrl set tx_enable 1
./phy_ctrl set tx_bs_mode 1
./phy_ctrl set rx_bs_mode 1
./phy_ctrl set ref1_auto_corr_thresh 125
./phy_ctrl set ref2_cross_corr_thresh 63
./phy_ctrl set ref2_auto_corr_thresh 188
./phy_ctrl set force_ref 1
./phy_ctrl set tx_gain_value 250
./phy_ctrl set tx_scaling -1
./phy_ctrl set tx_force_gain 1
./phy_ctrl set tx_frame 1
./phy_ctrl set rx_frame 1
./phy_ctrl set config_symb 1
./phy_ctrl set rx_buff_overflow 1
./phy_ctrl set cfg_symb_err 1
./phy_ctrl set int_header_err 1
./phy_ctrl set tx_fifo_reset 1
./phy_ctrl set rx_data_ready 0
./phy_ctrl set rx_dma_transf_width 1
./phy_ctrl set rx_dma 1
```

```
./phy_ctrl set tx_dma 1
./phy_ctrl set operate_mode 1
./phy_ctrl set tx_digital_iq_output 0
./phy_ctrl set rx_digital_iq_output 0
./phy_ctrl set analog_tx_sleep 1
./phy_ctrl set tx_freq_ratio 16384
./phy_ctrl set rx_freq_ratio 16384
./phy_ctrl set pdu_passthrough 0
./phy_ctrl set bad_data_filter 1
./phy_ctrl set fch_rate_id 6
./phy_ctrl set spi_pll_dac 128
./phy_ctrl set spi_tx_dac 90     #Setting for Anaconda 2
./phy_ctrl set spi_rx_dac 105     #Setting for Anaconda 2
./phy_ctrl set rx_noise_inducer 1
./phy_ctrl set base_id 9
./phy_ctrl set vga 81
insmod baseNetDrv.o
./sched_ctrl set cp_size 1/16
./sched_ctrl set channel_bw 3.5
./sched_ctrl set frame_code 2
#./sched_ctrl set duplex_mode fdd
./sched_ctrl set duplex_mode tdd
./sched_ctrl set FEC 1
./sched_ctrl set ttg 20
./sched_ctrl set rtg 20
./sched_ctrl set contention_slot 1
./sched_ctrl set feedback_mode none
./sched_ctrl set start 0
```

F.2 Subscriber Station Booting Script

To obtain the booting scripts type the following command at the Base Station, type:

`cat sub↵`

```
# cat sub
#/bin/sh
insmod anacondaDrv.o
insmod phyDriver.o
./anaconda_ctrl set sample_clk 40000000
./fpga_prog v2r0b71d6k.bin
./phy_ctrl set tx_mute_delay_lead 60
./phy_ctrl set tx_mute_delay_lag 4
./phy_ctrl set cp_size 16
./phy_ctrl set tx_enable 1
./phy_ctrl set tx_bs_mode 0
./phy_ctrl set rx_bs_mode 0
./phy_ctrl set ref1_auto_corr_thresh 125
./phy_ctrl set ref2_cross_corr_thresh 63
./phy_ctrl set ref2_auto_corr_thresh 188
./phy_ctrl set force_ref 1
./phy_ctrl set tx_gain_value 250
./phy_ctrl set tx_scaling -1
./phy_ctrl set tx_force_gain 1
./phy_ctrl set tx_frame 1
./phy_ctrl set rx_frame 1
./phy_ctrl set config_symb 1
./phy_ctrl set rx_buff_overflow 1
./phy_ctrl set cfg_symb_err 1
./phy_ctrl set int_header_err 1
./phy_ctrl set tx_fifo_reset 1
./phy_ctrl set rx_data_ready 1
./phy_ctrl set rx_dma_transf_width 1
./phy_ctrl set rx_dma 1
./phy_ctrl set tx_dma 1
./phy_ctrl set tx_digital_iq_output 0
```

```
./phy_ctrl set rx_digital_iq_output 0
./phy_ctrl set analog_tx_sleep 1
./phy_ctrl set tx_bw_mode 3.5
./phy_ctrl set rx_bw_mode 3.5
./phy_ctrl set tx_freq_ratio 16384
./phy_ctrl set rx_freq_ratio 16384
./phy_ctrl set digital_if_force_tx 1
./phy_ctrl set digital_if_force_rx 1
./phy_ctrl set spi_pll_dac 128
./phy_ctrl set spi_tx_dac 90       #Setting for Anaconda 2
./phy_ctrl set spi_rx_dac 105      #Setting for Anaconda 2
./phy_ctrl set pdu_passthrough 0
./phy_ctrl set bad_data_filter 1
./phy_ctrl set fch_rate_id 6       # 0=xxxa6k.bin, 8=xxxb6k.bin
./phy_ctrl set rx_noise_inducer 1
./phy_ctrl set base_id 9
./phy_ctrl set delay_correction 1001
./phy_ctrl set vga 81
insmod subNetDrv.o
./phy_ctrl set operate_mode 1
./ss_linktest_ctrl add 10 LinkTest
```

www.wavesat.com